5장. 배시쉘 활용하기(1)

Contents

학습목표

- 배시쉘이 제공하는 다양한 기능을 이해하고 활용한다.
- 환경변수를 이용해 작업의 효율을 높이는 환경을 설정하는 방법을 익힌다.
- 환경 설정 파일을 이용해 자신만의 환경을 설정하는 방법을 익힌다.

내용

- **쉘 기본 사용법 익히기**
- **입출력 방향 변경**
- 배시쉘의 환경 설정
- 명령 다루기
- 프롬프트 설정하기
- 환경 설정파일 설정하기

01. 기본 사용법 익히기

쉘 ?

- 사용자와 커널 사이의 중간 역할
- 사용자가 입력한 명령을 처리 하고 실행 결과를 알려줌
- 편리한 사용을 위해 다양한 기능 제공

쉘의 종류

- 본쉘(sh), C쉘(csh), 콘쉘(ksh), 배시쉘(bash),…

- 종류 확인하기
  - 프롬프트로 확인 : C쉘은 %, 본쉘/콘쉘/배시쉘은 $
  - 명령으로 확인

프롬프트 모양이 # 인 경우?

시스템관리자(root) 계정임

```
$ echo $SHELL
/usr/bin/bash
$
```

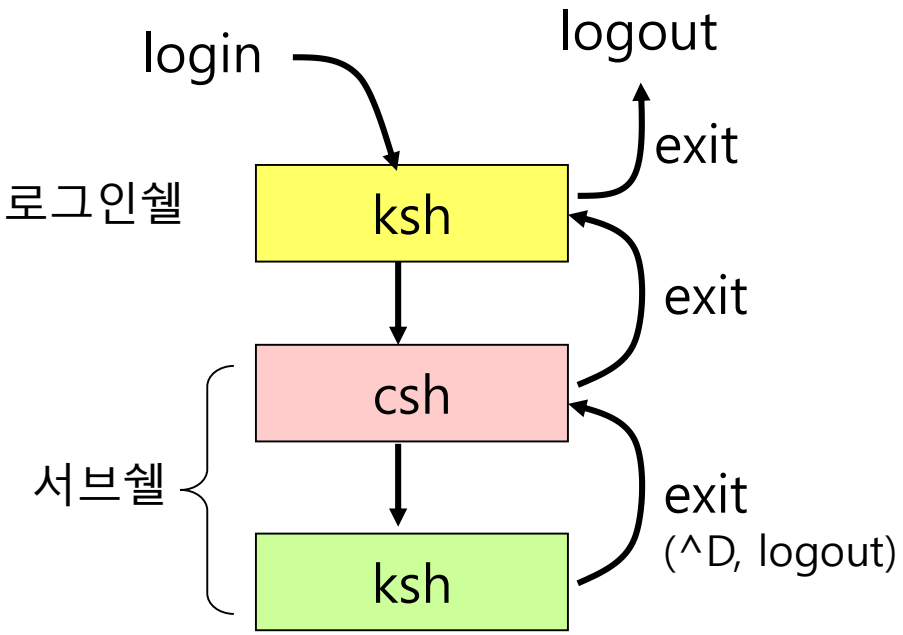
bash 쉘!

쉘 변경하기

로그인 쉘과 서브쉘

- 로그인 쉘 : 사용자가 로그인한 직후 자동 생성되는 쉘
- 서브쉘 : 사용자가 직접 실행한 쉘

```
$ csh
% ksh
$ exit
% exit
$
```



쉘(shell) 확인하기

- cat 명령어를 사용하여 /etc/shells 파일 내용 보기

```
[i20181234@localhost ~]$ cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/usr/bin/sh
/usr/bin/bash
/usr/sbin/nologin
/usr/bin/zsh
/bin/zsh
```

- echo 명령어를 사용하여 쉘(shell) 변수 확인

```
[i20181234@localhost ~]$ echo $SHELL
/bin/bash
```

문자열 출력

- **echo**
  - 쉘 명령과 유틸리티 모두 제공
  - 지정한 문자열 출력
- **print**
  - 쉘 명령
  - 지정한 문자열 출력

- **사용법**

```
$ echo test                  -> 쉘 내장 명령
test
$ /usr/bin/echo test         -> UNIX 유틸리티 실행
test
$ echo 'I love UNIX!!'
I love UNIX!!
$
```

쉘 특수문자

- 쉘이 특별하게 처리하는 문자들
- 잘 활용하면 명령을 입력할 때 매우 편리

메타문자	기능	예제
*	임의의 문자열	$ ls h*
?	임의의 한 문자	$ ls dir?
[ ]	대괄호안에 포함된 문자 중 하나	$ ls [a-f]*
~	홈 디렉토리	$ cd ~user1
-	이전 작업 디렉토리	$ cd -
;	명령 연결. 왼쪽부터 차례로 실행	$ date;cal;ls
|	왼쪽 명령의 실행 결과를 오른쪽 명령의 입력으로 전달	$ ls –l /etc | more
' '	모든 쉘 특수문자 무시	$ echo '$SHELL'
" "	$, `, ₩를 제외한 모든 쉘 문자 무시	$ echo "$SHELL"
` `	쉘 명령 수행	$ echo `date`
₩	특수문자 기능 제거	$ echo "₩$SHELL"
>, >> <, <<	입출력 방향 변경	

02. 입출력 방향 변경

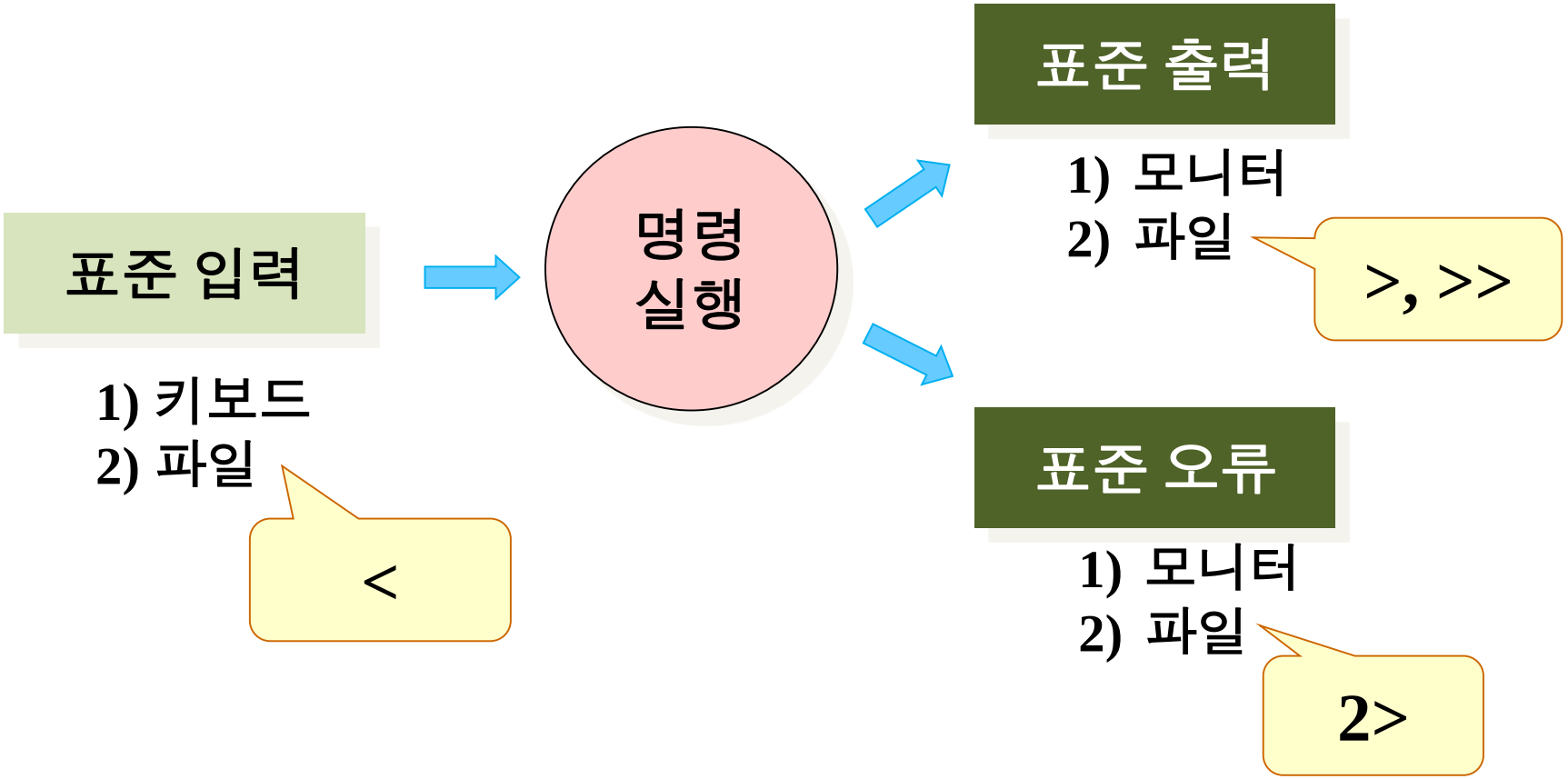
표준 입출력 장치

- 시스템이 명령의 입력과 출력에 사용하는 기본 장치

	파일 디스크립터	파일이름	설명	기본 장치
표준입력 Standard Input	0	stdin	명령의 실행에 필요한 데이터를 읽어들이는 기본 장치	키보드
표준출력 Standard Output	1	stdout	명령 실행 결과를 출력하는 장치	모니터
표준오류 Standard Error	2	stderr	명령 실행 중 발생한 오류 메시지를 출력하는 장치	모니터

리다이렉션 : 입출력 장치의 변경

- 특수 기호를 사용하여 표준 입출력 장치를 변경



출력 리다이렉션 : >, >>

$ 명령 > 파일명
$ 명령 >> 파일명

>는 원래 1>의 형태이나 1을 생략할 수 있음. 대부분 >을 사용함

- **표준출력을 모니터에서 파일로 변경**
  - > : 새로운 파일로 생성, 기존 파일의 내용은 없어짐
  - >> : 기존 파일의 끝에 내용 추가

- **사용법**

텔넷 cookook.co.kr

```
$ ls -al > test
$ date > test
$ cat test
Friday, May 18, 2012 12:30:16 AM KST
$ pwd >> test
$ cat test
```

결과는?

오류 리다이렉션

$ 명령 2> 파일명

- 표준 오류 메시지를 파일에 저장

- 사용법

```
$ ls  /ttt
/ttt: No such file or directory
$ ls /ttt 2> ls.err
$ cat ls.err
/ttt: No such file or directory
$
```

입력 리다이렉션

$ 명령 < 파일명

- 표준 입력을 키보드에서 파일로 변경
- 사용법

```
$ sort < test.sort
…

$ wall yoo pts/0 < test.in
```

다음 강좌에 계속