

1 ubuntu 환경 변수 vs. 지역 변수

-linux (ubuntu) 시스템에서 환경 변수와 지역 변수를 다루고 관리하는 법을 정리하겠다.

1. 환경 변수, 지역 변수란?

1)프로그래밍 언어를 이용해 코딩을 할 때 가장 많이 사용하는 것은 아마도 assignment 일 것이다.

-다음과 같이 변수에 값을 할당하는 것이다.

```
a = 1
```

2)환경 변수와 지역 변수 역시 위 코딩에서 a와 같은 변수이다.

-다만 운영 체제 내에서 쓰일 뿐이다.

-환경 변수와 지역 변수의 차이는 사실 그 이름에서도 나타난다.

3)환경 변수는 시스템 전체에 적용되고 있는 변수

4)지역 변수는 현재 터미널에서만 적용되는 변수

2. 환경 변수 다루기

1)이미 내 컴퓨터는 많은 환경 변수를 갖고 있다.

-기본적으로 시스템이 저장해두고 있거나 프로그램 설치 시 자동으로 추가되어있기 때문이다.

-현재 정의되어 있는 환경 변수 리스트를 보려면 다음 명령어를 입력한다

```
$ printenv
```

-내가 알고 싶은 환경 변수만 짚어서 보고 싶으면 그 이름을 같이 넣어주면 된다.

```
$ printenv HOME
```

-HOME이라는 환경 변수가 어떤 값을 갖고 있는 지 알 수 있다.

-그런데 이런 환경 변수들은 내가 직접 등록한 게 아니라서 환경 변수의 이름을 정확히 알고 있지 못하는 경우가 많다. 키워드로 찾아보고 싶을 때는 grep 을 해보자

```
$ printenv | grep "java"
```

2)위 명령을 통해 java라는 단어가 들어간 줄을 찾을 수 있어서 관련된 환경 변수의 이름과 그 값을 모두 볼 수 있다.

3)아까 printenv HOME을 통해 우리는 HOME이라는 환경 변수의 값이 무엇인지 확인했다.

-\$ 표시만 붙여주면 이 환경 변수를 시스템 안에서 자유롭게 사용할 수 있다.

-다음 명령어들을 통해서 간단히 체험해보자.

```
$ echo $HOME
```

```
$ cd $HOME/Documents
```

3. 지역 변수 다루기

1)현재 터미널에서 지역 변수를 생성해보자. 방법은 정말 간단하다.

```
$ LOCALVAR=VAL
```

2)LOCALVAR라는 변수를 만들어 그 안에 VAL이라는 값을 넣어두었다. 이게 제대로 잘 저장되어 있는지 확인해보자.

```
$ echo $LOCALVAR
```

3)VAL이라는 값이 튀어나오면 제대로 저장된 것이다.

-이제 방금 저장한 지역 변수 LOCALVAR가 환경 변수 리스트에 있는지 찾아볼까?

```
$ printenv LOCALVAR
```

```
$ printenv | grep "LOCALVAR"
```

4)위에서 두 변수의 차이점을 얘기했듯이, 환경 변수에 있을 리가 없다.

-또한 새로운 터미널을 열어서 echo \$LOCALVAR를 실행해보면 그런 변수가 없음을 알 수 있다.

-이를 통해 지역 변수는 해당 터미널에서만 작용하는 임시 변수임을 확인할 수 있다.

58 4. 환경 변수 등록 및 관리법

59 1)환경 변수를 등록하고 제거하는 가장 빠른 방법은 export와 unset을 직접 쓰는 것이다.

```
60  
61 # 환경 변수 등록  
62 $ export MY_ENV_VAR=VALUE  
63  
64 # 등록한 환경 변수 제거  
65 $ unset MY_ENV_VAR  
66
```

67 2)위 방법은 일반적으로 추천하지 않는 방법이다.

68 -환경 변수라는 것이 시스템을 사용하기 편리하도록 많은 주소들을 관리하는 목적으로 쓰는 것이 주 용도
중 하나인데, 이를 직접 다루다가 꼬이면 골치가 아파질 수 있다.
69 -또한 내가 직접 환경 변수를 등록해 놔다가 한 두 달 이따가 "그 때 환경 변수 뭐 했던 게 있었는데" 기억
못하고 그러면 난감하게 된다.

70
71 3)따라서 내가 직접 환경 변수를 조작하는 상황에서 필요한 것은? 기록하는 것이다.

72 -이미 ubuntu 시스템에서는 이러한 용도로 사용할 수 있는 /etc/environment 파일이 있다.
73 -찾아가서 열어보면 누가 써 놓은 것인지 모르지만 PATH라는 변수가 이미 정의되어 있는 것을 볼 수 있다.
74 -조금 더 자세히 살펴보면 내가 설치했던 프로그램들 중 일부가 추가해 놓은 것임을 알 수 있다.
75 -더 생각해보면 그 프로그램들이 스스로 "내가 PATH에 이렇게 값을 넣어놨소" 라고 사용자가 알 수
있도록 기록을 해 놓은 것임을 알 수 있다.
76 -혹시 그 프로그램을 삭제했는데 이 기록이 남아 있다면 확신을 가지고 지울 수 있을 것이다.
77

78 4)그럼 /etc/environment 파일에 내가 등록하고 싶은 환경 변수를 새로 한 줄 추가하고 저장하자.

79 -관리자 권한이 필요하므로 sudo를 이용해서 편집해야 할 것이다.
80 -그리고 source 명령을 통해 지역 변수로 올린다.

```
81  
82 $ source /etc/environment  
83  
84 # 새로 추가한 지역 변수 MYVAR를 확인하자  
85 $ echo $MYVAR  
86  
87 # 환경 변수에는 아직 없다  
88 $ printenv MYVAR  
89  
90 -이제 export를 통해 지역 변수를 환경 변수로 올려주면 된다.  
91 $ export MYVAR  
92  
93 # 환경 변수로 잘 등록되어 있는지 확인해보자  
94 $ printenv MYVAR  
95  
96  
97  
98
```

99 [Ubuntu] bash shell 환경변수

100 1. bash shell 환경변수

101 1)환경변수란?

102 -OS레벨에서 "자주사용하는" 또는 "필요한 변수들을 미리 선언해 놓은 것
103 -이 환경변수의 리스트를 보기 위해서는 export라는 명령어를 쓴다.
104 -1개의 환경변수를 보기 위해서는 echo \$환경변수(ex: echo \$PATH) \$PATH 환경변수에 있는
file들은 어느 위치에서나 실행 가능.

105

106

107 2. 환경변수 설정 방법

108 1)export 명령어를 통해 실행

```
109 export PATH=$PATH:$JAVA_HOME/bin  
110
```

111

112 2)하지만 이 방법은 재부팅 시엔 초기화됨

113

113

3).bashrc file 설정을 수정
vi ~/.bashrc 로 file을 연후

아래쪽에
export PATH="\$PATH:/usr/프로그램 경로/bin"
를 추가해준다

\$PATH: 를 앞에 추가해 준 것은 기존 PATH 환경변수 뒤에 내용을 추가한다는 의미

4)이부분을 빼먹으면 기존 설정은 삭제된다.

3. 환경변수 적용

1)컴퓨터 재시작을 하거나 또는

2)source .bashrc

와 같이 source 명령어를 이용하면 된다.