

## 패스워드 설정 확인 스크립트 설명

```
1 #!/bin/sh
2
3 expPeriod=`grep ^PASS_MAX_DAYS /etc/login.defs | awk '{print $2}'`
4 minDuration=`grep ^PASS_MIN_DAYS /etc/login.defs | awk '{print $2}'`
5 minLength=`grep minlen /etc/security/pwquality.conf | awk '{print $3}'`
6 expWarn=`grep ^PASS_WARN_AGE /etc/login.defs | awk '{print $2}'`
7
8 echo "*****"
9 echo "계정 패스워드 정책 확인"
10 echo "*****"
11 echo "패스워드 변경 주기 : $expPeriod일"
12 if [ $expPeriod -gt 90 ]; then
13 echo "패스워드 변경주기를 90일 이하로 변경해주세요\n"
14 else
15 echo "패스워드 변경 주기 정상\n"
16 fi
17 echo "패스워드 만료시 최소사용 기간 : $minDuration일"
18 if [ $minDuration -lt 1 ]; then
19 echo "패스워드 만료시 최소사용 기간을 1일 이상으로 변경해주세요\n"
20 else
21 echo "패스워드 만료시 최소사용기간 정상\n"
22 fi
23 echo "패스워드 최소 길이 : $minLength자"
24 if [ $minLength -lt 8 ]; then
25 echo "패스워드 최소 길이를 8자 이상으로 변경해주세요\n"
26 else
27 echo "패스워드 최소 길이 정상\n"
28 fi
29 echo "패스워드 만료 경고일 : $expWarn일"
30 if [ $expWarn -gt 30 ]; then
31 echo "패스워드 만료 경고일을 30일 이하로 변경해주세요"
32 else
33 echo "패스워드 만료 경고일 정상"
34 fi
```

nano check\_password.sh

1번 line : 인터프리터 선언 → /bin/sh 라는 프로그램을 통해 실행한다는 의미

3 ~ 6번 line : 일반적인 변수를 선언하는 과정 !!주의점!! 변수명=값 형식에서 중간에 공백이 있으면 안됨!!

값을 할당하는 과정에서 `` 문자로 묶여있다.(키보드 물결에 있는 문자) 이 문자는 해당 명령을 터미널에서 실행한 후 그 값을 반환한다.

eg) a=`echo "hi"` 라고 사용할 경우 a에는 hi가 저장됨

이 스크립트에서는 `grep <찾을 문자열> <찾을 파일명>` 형태로 사용한다. 여기서 `PASS` 앞에는 `^` 문자가 붙어있는데 이는 정규표현식에서 해당 문자로 시작하는 = 해당 문자가 맨 앞에 있는 내용을 찾은다는 것이다. eg) 파일에 `abc`와 `bc`가 있을 경우 “`grep bc 파일`”을 했을 때 `abc`의 `bc`가 먼저 있기 때문에 `abc`를 반환한다.

그러나 “`grep ^bc 파일`”을 했을 때는 `bc`로 시작해야하기 때문에 `abc`가 아닌 `bc`를 반환한다.

그 다음 `|` 문자의 경우(프로그래밍에서 OR로 사용하는 문자) ‘파이프’라는 개념의 문자이다. 이 파이프는 해당 문자를 기준으로 앞의 결과를 뒤 명령어로 전달한다.

마지막으로 `awk` 명령은 결과값에서 특정 값을 추출하기 위함이다. 기본적으로 `grep` 만 사용하였을 경우 아래와 같이 해당 문자열을 포함한 전체 행이 출력된다.

```
persian@Ubuntu:~$ grep ^PASS_MAX_DAYS /etc/login.defs
PASS_MAX_DAYS 99999
```

여기서 날짜값만 필요하기 때문에 `awk` 명령을 사용하여 추출하는 것이다. ‘`{print $2}`’는 결국 2라는 변수에 있는 값을 출력하라는 것인데, `awk`의 경우 공백을 기준으로 문자를 분할하기 때문에 여기서 2번 변수에 있는 값은 날짜가 된다. 이에 따르면 1번 변수에는

`PASS_MAX_DAYS`라는 값이 존재한다는 것을 알 수 있다.

`echo`의 경우 터미널에 출력하는 명령이므로 생략하고 `if` 문의 형태는 다음과 같이 요약할 수 있다.

```
if [ 조건 ]; then
    실행할 명령
else
    실행할 명령
fi
```

여기서 중요한 것은 조건을 적는 대괄호에서 꼭 앞뒤로 공백을 한 칸씩 줘야 한다는 것이다.

조건문을 보면 `[ 변수 ( -gt | -lt ) 숫자 ]`의 형태를 가지고 있는데 이는 변수의 값이 숫자보다 큰지 혹은 작은지 비교하는 것이다.

`-gt`의 경우 greater than 즉, 보다 클 경우이며 lower than은 반대로 보다 작을 때를 의미한다.