

1. ALTIBASE 운영을 위한 Linux 설정 가이드

2

1.1 1. 커널파라미터

2

1.2 2.사용자 환경

6

1.3 3. 요약

8

# ALTIBASE 운영을 위한 Linux 설정 가이드

## 개요

본 문서는 리눅스(Linux) 운영체제(Operating System)에서 ALTIBASE를 설치하고 운영하기 위한 커널파라미터(kernel parameter)의 적절한 설정 값 및 사용자 환경변수에 대한 가이드를 제시합니다

본 문서에서는 ALTIBASE가 설치되기 전 설정해야 할 운영체제관련 사항에 대해서만 가이드를 제시하며 ALTIBASE 자체 설정을 위한 ALTIBASE 프로퍼티(property) 설정파일(altibase.properties)에 관해서는 별도문서 『ALTIBASE 설정파일 가이드』를 참조하시기 바랍니다.

본 문서와 관련된 오류 및 개선사항은 <http://support.altibase.com> 로 문의 주시기 바랍니다.

대표번호 02-2082-1000 , 기술 지원 센터 02-2082-1114

## 1. 커널파라미터

- 커널파라미터
  - 공유메모리
    - 리눅스에서의 설정
    - 메모리 DB를 공유메모리에 적재할 경우 고려사항
  - 세마포어
    - 리눅스에서의 설정
    - 유의사항
  - 파일캐시
    - 리눅스에서의 설정
  - THP(Transparent Huge Page)
    - 리눅스에서의 설정
  - 설정예제
    - 공유메모리
    - 세마포어
    - 공유메모리 및 세마포어 확인방법
    - 파일캐시
    - THP(Transparent Huge Page)

## 커널파라미터

리눅스에서 운영체제에서 ALTIBASE를 설치하고 운영하기 위해 설정해야 할 커널파라미터의 종류와 어떤 이유로 설정을 변경해야 하는지에 대해 설명한다. 마지막 섹션에서는 설정예제를 소개한다.

각 커널파라미터 및 설정방법과 관련된 자세한 사항은 리눅스에서 제공하는 가이드를 참조하도록 한다

## 공유메모리

응용프로그램을 개발하는 경우 두 개 이상의 프로세스가 정보를 주고 받아야 할 필요성이 발생한다. 운영체제는 이를 위해 IPC (Inter Process Communication)라고 부르는 자원을 제공하고 있다. 여러 IPC자원 중에 두 개 이상의 프로세스가 정보를 주고 받기 위해 사용되는 메모리영역을 공유메모리(shared memory)라고 부른다. 공유메모리는 사용자가 단위를 지정하여 한 개 이상의 영역으로 나누어 설정할 수 있다. 한 개의 공유메모리 영역을 세그먼트(segment)라고 부른다.

예를 들어 사용자는 10 MB로 1개의 세그먼트를 갖는 공유메모리를 설정할 수도 있고 10 MB영역을 10개의 세그먼트로 구성하여 100 MB의 공유메모리를 설정할 수도 있다. ALTIBASE 는 기본설정으로 메모리 DB를 프로세스의 힙(heap)영역에 적재하지만 사용자의 설정에 따라서 공유메모리에 적재하여 사용할 수 있다.

혹은, DBMS와 접속하는 클라이언트의 통신버퍼를 공유메모리를 통해 주고 받을 수 있도록 설정할 수 있다. (이와 관련된 설정방법은 『ALTIBASE 설정파일 가이드』 문서를 참고하도록 한다) 따라서, 세그먼트의 최대크기나 개수 제한과 같은 공유메모리 사용과 관련된 커널파라미터 설정이 사전에 필요하다.일반적인 유닉스 시스템에서 공유메모리 관련 커널파라미터와 권장 값은 아래와 같다

커널파라미터	설명	권장 값
shmmax	공유메모리 세그먼트 하나의 최대크기	2G+1
shmmni	시스템내의 공유메모리 세그먼트의 최대개수	500이상
shmseg	하나의 프로세스가 최대로 소유할 수 있는 공유메모리 세그먼트의 개수	200이상

리눅스에서는 추가적으로 shmall이 존재한다.

커널파라미터	설명	권장 값
shmall	시스템내의 공유메모리 세그먼트의 최대합계	물리메모리의 90%

## 리눅스에서의 설정

- shmmax는 권장 값인 2 G로 변경하는 것을 권장한다. 참고로 x86 시스템에서 필요한 최소값은 268435456 바이트(256 MB)이고 64비트 시스템의 경우에는 1073741824 바이트(1 GB)이다.
- shmmni는 기본값이 4096이므로 특별히 변경하지 않아도 된다.
- shmseg은 하드코딩 되어 있어 변경자체가 불가능하므로 무시하여도 된다.
- shmall은 8 GB 이상의 메모리 DB를 공유메모리에 적재 방식으로 운영할 경우에만 설정한다.

## 메모리 DB를 공유메모리에 적재할 경우 고려사항

메모리 DB를 공유메모리에 적재할 경우에는 아래 사항들을 추가로 고려해야 한다.

- shmmax  
shmmax의 경우는 구동단계에서는 ALTIBASE 설정파일 내에 STARTUP\_SHM\_CHUNK\_SIZE 항목의 크기만큼 한 개의 세그먼트가 만들어지기 때문에 STARTUP\_SHM\_CHUNK\_SIZE 보다 큰 값으로 설정하도록 한다.
- shmmni  
메모리 DB의 사용량이 증가하게 되면 ALTIBASE는 자동적으로 메모리영역을 증가시킨다. 이 경우 역시 세그먼트가 새롭게 생성되는데 이 세그먼트의 크기는 ALTIBASE설정파일 내에 EXPAND\_CHUNK\_PAGE\_COUNT \* 32K 단위로 세그먼트가 생성되기 때문에 shmmni는 가능한 여유 있게 설정하도록 한다.
- shmall  
메모리 DB의 최대크기인 MEM\_MAX\_DB\_SIZE에 따라 shmall을 조정해야만 한다. 상세한 내용은 [설정예제] 섹션에서 shmall을 참조하는 것으로 한다.

## 세마포어

공유메모리와 동일하게 IPC자원의 일종이다. 차이점은 공유메모리가 데이터를 프로세스 간에 공유하기 위한 방법으로 제공되는 반면에 세마포어는 프로세스 간의 동기화를 구현하기 위한 방법으로 운영체제에서 제공되는 자원이다. 동기화란 특정자원이나 객체에 대해 한 시점에 하나의 프로세스만이 접근하여 변경 가능하도록 제어 하는 것을 의미한다. (유크스와 다르게 조화는 동시에 가능하게 사용할 수 있다.)

ALTIBASE는 앞에서 설명한대로 로컬서버 내에 ALTIBASE 서버와 클라이언트가 존재할 경우 양단간의 통신버퍼로 공유메모리를 사용하는데 이때, 메모리에 대한 동시성을 제어해야 할 필요가 있다. (동시에 쓰기를 진행하면 안되기 때문이다.)

이 경우 메모리에 대한 읽기/쓰기에 대한 접근권한의 제어를 위해 세마포어 오퍼레이션(semaphore operation)이 발생하게 된다.

세마포어 오퍼레이션에 따라 프로세스는 대기상태가 되거나 혹은 진행상태가 될 수 있다. 이러한 세마포어 오퍼레이션은 동시다발적으로 발생되기 때문에 세마포어의 개수 및 오퍼레이션에 대한 적합한 커널파라미터 설정이 필요하다.

일반적인 유닉스 시스템에서 세마포어 관련 커널파라미터와 권장 값은 아래와 같다.

커널파라미터	설명	권장 값
semmns	시스템 내 세마포어의 최대개수이며 16바이트의 커널메모리가 할당된다.	8192 이상
semmni	세마포어 세트의 최대개수로 65535 이내에 설정가능하며 하나의 세트당 84바이트의 커널메모리가 할당된다.	5029
semmsl	한 개의 세마포어 세트에 존재하는 세마포어의 최대개수이며 논리적으로 semmns와 같거나 작아야 한다. 너무 크게 잡으면 몇 개의 세마포어 ID가 시스템 전체의 세마포어를 독점할 수 있다.	2000
semmap	semget 호출에 의해 얻어지는 세마포어 공간을 의미	5024
semmnu	시스템 내의 undo structure의 최대개수	1024
semopm	semop 시스템 콜에서 처리하는 operation의 최대개수	512
semume	한 개의 프로세스당 undo entry의 최대개수	512
semvmx	한 개의 세마포어의 최대값을 제한하며 32767보다 크게 지정하지 않는다.	32767

## 리눅스에서의 설정

이전에 별도의 설정이 없었다면 세마포어 관련 커널파라미터로 semmsl, semmns, semopm, semmni 만 설정해도 무방하다. 이 중에서 semmns는 기본값이 권장 값보다 큰 32000이므로 변경하지 않아도 되나 명령문 특성상 함께 기술하여야 한다.

## 유의사항

Linux Kernel 2.5 이하의 커널 버전에서는 쓰레드 프로그램에서 SEM\_UNDO를 사용하는 문제점이 보고되어 있다.

 <http://www.uwsg.iu.edu/hypermil/linux/kernel/0202.0/0819.html>

이러한 문제점은 클라이언트의 빈번한 연결 및 해제가 다른 클라이언트의 동작에 영향을 미쳐 정상으로 동작하는 클라이언트의 접속이 끊어지는 현상을 발생시킬 수 있다. 따라서, Linux Kernel 2.5 이상을 사용하는 것을 권장한다.

커널 버전을 확인하는 명령어는 아래와 같다.

```
# uname -r
```

## 파일캐시

본 커널파라미터는 변경이 필수적인 요소는 아니나, 적절한 파일캐시(file cache) 설정으로 ALTIBASE가 사용하는 메모리 영역에 대한 스왑아웃(swap out) 발생요건을 억제, 스왑핑(swapping)으로 인한 운영체제 계층의 Disk I/O 지연시간이 ALTIBASE의 성능저하로 이어지는 현상을 최소화하기 위해 권고하는 사항이다.

파일캐시란 주기억장치와 보조기억장치간의 속도차이로 인한 병목을 해소하기 위해 운영체제차원에서 관리되는 일종의 시스템버퍼이다. 이러한 파일캐시는 운영체제 저마다의 고유한 정책(policy)에 의하여 관리되나 공통적으로 스왑정책과 직접적인 상관관계를 가진다.

스왑핑이란 자체는 주기억장치보다 더 큰 응용프로그램이나 데이터 파일을 다룰 수 있게 하는 유용함을 가지고 있으나, DBMS와 같이 장기상주형의 응용프로그램이 운영되는 시스템에서는 스왑핑으로 인한 운영체제 계층의 Disk I/O 지연시간으로 인해 DBMS의 응답시간이 불규칙해지거나 최악의 경우에는 행(hang)과 같은 현상이 발생할 수 있기에 시스템 용도에 따라 고려요소가 된다.

따라서, ALTIBASE의 일관적인 응답시간 보장을 위해서는 스왑이 최대한 발생하지 않도록 관련 파일캐시 및 스왑 커널들을 사전에 설정하는 것이 바람직하다.

## 리눅스에서의 설정

리눅스의 경우는 파일캐시 관련 커널파라미터를 조정하는 방법도 있지만 아래 커널파라미터를 통해 스왑 발생에 대한 간접적인 제한이 가능하다

커널파라미터	설명	권장 값
swappiness	스왑 성향을 0-100의 수치로 정의한다. 값이 작을수록 적은 양의 스왑이 발생한다. 기본값은 60이다.	30 이하

## THP(Transparent Huge Page)

RedHat 6.0부터 추가된 기능으로 커널에서 메모리 관리(페이지단위)를 기존 4K 단위에서 2M라는 큰 페이지로 관리한다. 메모리를 사용하는 APP 특성에 따라 장단점이 있으나 인메모리 DBMS 에서는 사용하지 않는 것을 권장한다.

☐ THP 에 관한 설명 참고: <https://access.redhat.com/solutions/46111>

THP 설정이 Enable(on) 되어 있는지 여부는 아래의 명령어 확인할 수 있다.

- THP 설정 상태 확인

```
$ cat /sys/kernel/mm/transparent_hugepage/enabled
[always] madvise never
```

위의 결과처럼 always 로 출력될 경우 THP 설정이 Enable(on) 되어 있는 상태이다.

## 리눅스에서의 설정

THP 설정이 Enable 되어 있을 경우 메모리 할당 지연 현상이 발생해서 급격한 성능저하가 발생하는 사례가 다수 발견되었다. 따라서 THP 설정을 사용하지 않음으로 설정을 변경할 것을 권고한다.

커널파라미터	설명	권장 값
THP(transparent_hugepage)	커널에서 메모리 관리 페이지 단위를 기존 4K에서 2M라는 단위로 변경한다. 기본값은 Enable이다.	never(Disable 의미)

## 설정에제

리눅스에서 커널파라미터를 설정하는 방법은 크게 세 가지로 Redhat 6.2부터 지원되는 커널파라미터 관련 유틸리티인 sysctl을 수행하는 방법, /etc/sysctl.conf 파일을 직접 변경하는 방법, 구동 중에만 유효한 커널파라미터 변경 명령어를 시스템 구동 시 자동으로 수행하는 쉘인 /etc/rc.d/rc.local 에 추가하는 방법이 있다. 여기서는 세 번째 방식으로 설정하는 예를 다룬다.

기본적으로 root 계정으로의 접속을 필요로 하며 올바르게 적용하기 위해서는 변경 후 시스템을 재 구동을 하여야만 한다. 참고로 두 가지 변경 방법을 모두 적용 시는 최종적으로 /etc/rc.d/rc.local이 반영되므로 /etc/sysctl.conf 파일을 통한 설정이 무시된다는 것을 유의해야 한다

## 공유메모리

아래 각 명령어를 /etc/rc.d/rc.local 에 추가한다. 단, shmall은 앞서 설명하였듯이 8GB이상의 공유메모리 적재 방식으로 운영할 경우에만 설정하도록 한다.

- shmmax

```
echo 2147483648 > /proc/sys/kernel/shmmax
```

- shmmni

```
echo 4096 > /proc/sys/kernel/shmmni
```

- shmall

shmall은 기본값이 8 GB이다. 따라서, 물리 메모리가 8 GB를 초과하는 시스템에서 공유메모리를 최대한으로 사용하려면, 이 값을 물리 메모리의 약 90%로 늘려야만 한다. 이때, 값의 단위는 리눅스 메모리 페이지(memory page) 개수이다.

예를 들어, 시스템의 물리 메모리가 16 GB인 경우, 16 GB의 90%는 14.4 GB이고 이를 리눅스 페이지 기본 크기인 4 KB로 나누면 3774873이 되므로 아래와 같은 명령문이 된다.

```
echo 3774873 > /proc/sys/kernel/shmall
```

참고로 리눅스 메모리 페이지 크기는 4KB 이나, 정확한 사이즈를 확인하기 위해서는 다음 명령어를 수행하도록 한다.

```
# getconf PAGE_SIZE
```

## 세마포어

아래 명령어를 /etc/rc.d/rc.local 에 추가한다.

```
echo 2000 32000 512 5029 > /proc/sys/kernel/sem
```

## 공유메모리 및 세마포어 확인방법

설정된 값은 아래와 같은 방법으로 확인이 가능하다.

```
----- Shared Memory Limits -----
max number of segments = 4096           //SHMMNI
max seg size (kbytes) = 2097152         //SHMMAX
max total shared memory (kbytes) = 15097856 //SHMALL
min seg size (bytes) = 1                //SHMMIN

----- Semaphore Limits -----
max number of arrays = 5029             //SEMMNI
max semaphores per array = 2000         //SEMMSL
max semaphores system wide = 32000      //SEMMNS
max ops per semop call = 512            //SEMOPM
semaphore max value = 32767             //SEMMVMX

----- Messages: Limits -----
max queues system wide = 16
max size of message (bytes) = 8192
default max size of queue (bytes) = 16384
```

## 파일캐시

아래 명령어를 /etc/rc.d/rc.local 에 추가한다.

```
echo 30 > /proc/sys/vm/swappiness
```

## THP(Transparent Huge Page)

root 의 권한을 가지고 아래의 명령어를 순차적으로 실행하여 THP 설정을 Disable 시킨다.

- 설정 Disable 하기

```
echo 30 > /proc/sys/vm/swappiness
```

- system reboot 후에도 지속적으로 disable 속성을 유지하기 위해서 아래의 명령어를 /etc/rc.d/rc.local 에 추가한다.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

## 2.사용자 환경

### 사용자환경

리눅스 운영체제에서 ALTIBASE 를 설치하고 운영하기 위해 설정해야 할 사용자 환경의 종류와 어떤 이유로 설정을 변경해야 하는지에 대해 설명한다. 마지막 섹션에서는 설정예제를 소개한다.

각 사용자 환경 및 설정방법과 관련된 보다 자세한 사항은 리눅스에서 제공하는 가이드를 참조하도록 한다

### 리소스 제한

유닉스 운영체제에서는 사용자계정 단위로 사용 가능한 리소스에 대한 논리적인 제한 값을 설정한다. 이 리소스 제한 항목 중, 안정적인 서비스 운영을 위해 확장시켜야 하는 일반적인 유닉스 운영체제에서의 항목들은 아래와 같다.

항목	설명	권장 값
data seg size(data)	프로세스 데이터영역의 최대크기	unlimited
file size (fsize)	생성 가능한 파일의 최대크기	unlimited
open files (nofiles)	동시에 접근 가능한 파일의 최대개수	unlimited
max memory size (rss)	사용 가능한 메모리의 최대크기	unlimited
virtual memory (memory)	사용 가능한 가상메모리의 최대크기	unlimited
max user process	프로세스의 최대개수	unlimited

어디까지나 특정 사용자가 사용하는 메모리 및 데이터파일 영역 확장 시 물리적인 리소스가 풍부한데도 논리적인 제한으로 인해 발생할 수 있는 문제를 사전에 제거하기 위한 것으로, 이 설정으로 인해 다른 프로세스에게 끼치는 영향은 없기에 해당 운영체제에서 허용하는 최대값(가능하다면 unlimited)으로 설정하는 것을 권고한다.

예를 들어, open files의 의미는 프로세스가 접근하는 파일뿐만 아니라 통신소켓의 개수까지 포함하고 있다. 만일, 이 값이 10으로 제한된 환경에서 ALTIBASE 를 운영할 경우 동시에 접속 가능한 세션은 10개 이상은 불가능하다는 의미이다. (실제로 ALTIBASE가 사용하는 파일까지 고려하면 접속 가능한 세션이 없을 수도 있다.)

변경방법은 ulimit 명령을 사용한 환경설정파일 편집 및 시스템 리소스 설정파일 편집 또는 해당 운영체제 별로 제공하는 커널관련 유틸리티 활용이 있다.

참고로 리눅스의 경우 open files가 unlimited로 지정되지 않으므로 최대값인 1048576로 설정한다.

### Hard-Limit & Soft-Limit

리소스 제한 값은 hard-limit, soft-limit 이라는 개념으로 구분된다. hard-limit은 시스템어드민계정(root)외에는 변경할 수 없는 커널차원의 리소스 제한 최대값을 의미하며 soft-limit은 현재 사용자계정 내에서는 hard-limit까지 변경이 가능하다는 것을 의미한다. (상세한 내용은 ulimit -S / -H 옵션을 참고한다.)

soft-limit은 사용자가 접속을 하여 세션을 유지하는 동안 유효하며 변경을 하면 즉시 반영된다. 하지만 동일 사용자계정의 다른 세션이 연결되면 기존의 soft-limit이 반영되기 때문에 편의상 ulimit 명령어를 사용자계정의 환경설정파일에 추가하곤 한다.

하지만, 이러한 방법은 전역적인 hard-limit에 의해 의도한 설정이 되지 않을 수 있으므로 환경설정파일을 이용한 사용자계정 단위 적용보다는 시스템 차원의 리소스 설정파일 편집을 통하여 체계적으로 적용하는 것을 권고한다

참고로 리눅스에서 사용자 리소스 제한관련 시스템 설정파일은 /etc/security/limits.conf 이다.

### 환경변수

필수적인 환경변수는 아래와 같다. 아래 환경변수를 설정하고자 하는 사용자의 셸(shell)의 문법(syntax)에 따라 사용자계정의 환경설정파일에 반드시 설정하도록 한다.

환경변수	설명
ALTIBASE_HOME	ALTIBASE를 설치한 경로를 지정한다.

PATH	ALTIBASE_HOME/bin 에 있는 바이너리 및 쉘 스크립트를 경로에 상관없이 활용하기 위해 경로를 지정하여 준다.
LD_LIBRARY_PATH	ALTIBASE의 동적 라이브러리를 사용할 경우 라이브러리가 존재하는 경로를 지정하여 준다.

## 문자세트 고려사항

ALTIBASE 5.3 버전 이전에는 문자세트와 관련하여 NLS\_USE 라는 항목과 관련해 별도의 프로퍼티가 존재 하였으나 ALTIBASE 5.3 이후로 환경변수로 대체되었다.

이로 인해, ALTIBASE 5.3 버전 이상에서 DB의 문자세트를 US7ASCII가 아닌 KO16KSC5601 / MS949 / UTF8 등을 설정했을 경우에는 반드시 ALTIBASE\_NLS\_USE 라는 환경변수를 설정해야만 한다. 그렇지 않을 경우에는 기본설정인 US7ASCII로 접속하게 되어 입력 후 조회 시점에 알 수 없는 문자열 형태로 조회가 되게 된다.

## 터미널 환경

[[환경변수](#)] 섹션에서 언급한 문자세트 사용관련 고려사항과 더불어, 사용하는 터미널의 폰트(font) 역시 DB에 설정한 문자세트와 동일하게 설정하여야 한다.

## 설정예제

sh, bash, ksh에서 환경설정파일을 이용한 환경변수 및 리소스 제한을 설정하는 예제이다. csh의 경우는 export 대신 setenv 와 같은 쉘 명령어를 통해 선언한다.

```
#For ALTIBASE
ALTIBASE_HOME=/ALTIBASE: export ALTIBASE_HOME;
PATH=$ALTIBASE_HOME/bin:$PATH: export PATH;
LD_LIBRARY_PATH=$ALTIBASE_HOME/lib:$LD_LIBRARY_PATH;
export LD_LIBRARY_PATH;
ALTIBASE_NLS_USE=MS949: export ALTIBASE_NLS_USE;

#For Resource Limitation
ulimit -d unlimited #data segment size
ulimit -f unlimited #file size
ulimit -n 1048576 #file descriptor(open files)
ulimit -m unlimited #max memory size(rss)
ulimit -v unlimited #virtual memory
ulimit -u unlimited #user process
```

참고로 ksh의 경우, 환경변수가 미리 정의되지 않은 상태에서 환경변수를 사용하여 또 다른 환경변수를 정의 시 오류가 발생할 수 있다.

## 3. 요약

### 요약

리눅스 운영체제에서 ALTIBASE의 안정적인 운영을 위해서는 커널파라미터 설정 및 사용자 환경 설정을 사전에 수행해야만 한다. 설정을 올바르게 수행하지 않은 경우, 시스템의 리소스가 충분함에도 불구하고 각각의 제한 값들에 의해 문제가 발생 할 수 있다.

설정방법은 각 챕터의 [설정예제] 섹션을 참조하도록 한다.

## 커널파라미터설정

아래 표를 참조, 적절한 커널파라미터를 설정하도록 한다.



분류	커널파라미터	권장 값	비고
공유메모리	shmmax	2G+1	–
	shmmni	500이상	기본값이 4096이므로 확인으로 대체.
	shmall	물리 메모리의 90%	8GB 이상의 공유메모리 적재 방식일 경우에만 설정 필요.
세마포어	semmsl	2000	너무 큰 값을 설정하지 않도록 주의.
	semmns	8192이상	기본값이 32000이므로 확인으로 대체.
	semmni	5029	–
	semopm	512	–
파일캐시	swappiness	30 이하	–
THP(Transparent Huge Page)	transparent_hugepage	never	사용안함설정으로 변경

공유메모리에 적재 방식으로 운영할 경우는 추가적으로 고려하여야 하는 요소가 있다. [공유메모리] 섹션을 참조하도록 한다.

## 사용자환경 설정

### 사용자 리소스 제한

아래 표를 참조, 가능하다면 모두 무제한으로 설정하도록 한다.

항목	설명	권장 값
data seg size(data)	프로세스 데이터영역의 최대크기	unlimited
file size (fsize)	생성 가능한 파일의 최대크기	unlimited
open files (nofiles)	동시에 접근 가능한 파일의 최대개수	1048576
max memory size (rss)	사용 가능한 메모리의 최대크기	unlimited
virtual memory (memory)	사용 가능한 가상메모리의 최대크기	unlimited
max user process	프로세스의 최대개수	unlimited

### 사용자 환경변수

아래 표를 참조, 사용자 환경변수를 설정하도록 한다.

환경변수	설명
ALTIBASE_HOME	ALTIBASE 설치 경로로 선언
PATH	\$ALTIBASE_HOME/bin 추가
LD_LIBRARY_PATH	\$ALTIBASE_HOME/lib 추가
ALTIBASE_NLS_USE	ALTIBASE 의 문자세트로 선언