

Real Alternative DBMS ALTIBASE, Since 1999

## **Altibase 운영을 위한 Linux 설정 가이드**



---

## 목차

개요 .....	3
LINUX 호환성 .....	4
glibc 호환성 버전.....	4
glibc 권고 버전.....	4
glibc 버전 확인 방법.....	4
커널 파라미터 .....	5
CPU frequency Governor.....	5
RemoveIPC.....	5
swappiness.....	6
THP(Transparent Huge Pages) .....	7
max_map_count .....	7
공유 메모리(Shared memory) .....	7
세마포어(Semaphore).....	8
커널 파라미터 변경 방법 .....	10
CPU frequency Governor .....	10
RemoveIPC.....	13
swappiness.....	14
THP.....	16
max_map_count .....	19
공유 메모리와 세마포어.....	21
사용자 환경 .....	23
리소스 제한.....	23
환경변수.....	25
요약 .....	28
기타 .....	29
Red Hat Enterprise Linux 권장 Swap 크기.....	29
참고자료.....	30
Linux 호환성.....	30
CPU frequency Governor.....	30
RemoveIPC.....	30
swappiness.....	30
THP.....	30
max_map_count .....	31
공유 메모리 및 세마포어.....	31
로케일(Locale) .....	31
MALLOC_ARENA_MAX.....	31

---

## 개요

이 문서는 Linux 시스템에서 Altibase 서버를 안정적으로 운영하기 위한 커널 파라미터 및 OS 사용자 환경설정에 대한 가이드를 제시한다.

이 문서는 아래 버전을 기준으로 작성되었다.

- Altibase 5.5.1 이상
- Red Hat Enterprise Linux 6 이상

이 문서와 관련된 오류 및 개선사항은 기술지원포털 또는 기술지원센터로 문의주시기 바랍니다.

- 기술지원포털 : <http://support.altibase.com> ▶ 기술 지식 ▶ Q&A
- 기술지원센터 : 02-2082-1114

---

## Linux 호환성

Linux는 수많은 배포판이 존재하나 Altibase와의 호환성 확인은 배포판 종류 및 커널 버전에 상관없이 glibc 버전을 기준으로 한다.

---

### glibc 호환성 버전

- Altibase 5.5.1 부터 glibc 버전을 기준으로 호환성 여부를 확인한다.
- Altibase 서버 버전 별 호환성이 보장된 glibc 버전은 아래와 같다.

Altibase 서버 버전	glibc 버전
Altibase 7.1.0	2.12 ~ 2.20
Altibase 6.5.1	2.12 ~ 2.20
Altibase 6.3.1	2.3.4 ~ 2.20
Altibase 6.1.1	2.3.4 ~ 2.20
Altibase 5.5.1	2.3.4 ~ 2.20

---

### glibc 권고 버전

- glibc-2.12-1.166.el6\_7.1 이상을 권고한다.
- 이전 버전의 glibc는 시스템 콜(malloc/free) 함수가 race condition으로 인해 deadlock이 발생할수 있는 버그가 존재한다.

---

### glibc 버전 확인 방법

- glibc 버전 확인 방법은 다음과 같다.

```
$ rpm -q glibc
```

---

## 커널 파라미터

Linux에서 Altibase를 설치하고 안정적으로 운영하기 위해 설정해야 할 커널 파라미터의 종류와 권고값, 권고 이유에 관해 설명한다.

설정 권고 파라미터 종류는 다음과 같다.

- CPU frequency Governor
- RemoveIPC
- swappiness
- THP
- max\_map\_count
- 공유 메모리
- 세마포어

---

### CPU frequency Governor

Altibase 서버는 최대 처리 성능과 최단 응답시간을 절대적으로 요구하는 시스템이므로 CPU 클럭 속도를 항상 최고로 유지해야 한다.

Linux는 그린 IT(환경친화적인 컴퓨팅)에 발맞춰 시스템의 전력 소모를 효율적으로 최소화하는 것이 전원 관리의 핵심으로 보고 이를 위해 CPUfreq Governor를 제공한다.

RHEL 6의 기본 설정 Ondemand Governor는 주파수 변경에 따른 지연 시간으로 Altibase 서버 성능이 일관성을 보장하지 못한 사례가 있다. 따라서 performance로 설정하거나 이 기능 자체를 비활성화하는 것을 권고한다.

커널 파라미터	설명	권고값
CPU frequency Governor (CPUfreqGovernor)	Linux에서 전원 관리를 위해 CPU 주파수 변경 규칙 등을 조정하는 성능 조정기이다. RHEL 6의 기본값은 Ondemand로 시스템의 부하가 높을 때는 CPU가 최고 클럭 주파수로 동작하고, 시스템이 유휴상태 일 때는 최저 주파수로 동작한다.	performance

#### Linux 커널 버전

- CPUfreq Governor는 Red Hat Enterprise Linux 6 부터 지원하는 기능이다.
- RHEL 6의 커널 버전은 2.6.32-71 이다.
- [Red Hat Enterprise Linux Release Dates](#)

---

### RemoveIPC

RemoveIPC는 RHEL 7.2에서 추가된 옵션으로 OS 사용자가 세션을 종료할 때 System V IPC와 POSIX IPC 객체를 제거하는 리눅스 프로퍼티이다.

기본 설정 yes의 경우 IPC를 사용하는 Altibase 환경에서 강제적인 세마포어 할당, 반환으로 Altibase 서버 및 애플리케이션 비정상 종료 현상이 발생한 사례가 보고되었다. IPC를 사용하지 않는 환경에서도 커널에서 발생하는 RemoveIPC 관련 동작 및 시스템 콜로 인해 시스템 성능에 영향을 주는 사례도 보고되었으므로 기본적으로 no를 권고한다.

커널 파라미터	설명	권고값
RemoveIPC	OS 사용자가 로그아웃할 때 IPC 자원을 모두 제거한다. root 사용자와 시스템 사용자는 이 설정에 영향을 받지 않는다. 기본값은 yes이다.	no

#### Linux 커널 버전

- RemoveIPC는 Red Hat Enterprise Linux 7.2 에서 추가된 옵션이다.
- RHEL 7.2 의 커널 버전은 3.10.0-327 이다.
- [Red Hat Enterprise Linux Release Dates](#)

## swappiness

스와핑(swapping) 발생 시 디스크 I/O로 인한 시스템 성능 저하가 Altibase 서버 성능에 영향을 미치는 것을 최소화하기 위해 권장하는 커널 파라미터이다.

스와핑이란 물리 메모리가 부족한 상황에서 사용 빈도가 낮은 물리 메모리 영역을 스왑 영역으로 내리는 작업(스왑 아웃, swap out)을 의미한다. 스왑은 디스크를 메모리처럼 사용하기 위한 것으로 스왑 아웃이 발생하면 DISK I/O로 인한 시스템 성능 저하가 발생한다.

따라서 Altibase 서버의 안정적이고 일관적인 성능을 유지하기 위해 swappiness를 적절하게 설정하는 것이 중요하다.

커널 파라미터	설명	권고값
swappiness	0 ~ 100 사이에서 설정할 수 있다. 낮은 값은 커널이 가급적 페이지 캐시의 페이지를 사용하며 높은 값은 물리 메모리에서 사용 빈도가 낮은(cold) 페이지를 스왑 아웃하는 것을 선호한다.	1

- swappiness를 완전히 비활성화하면 메모리 부족 상황에서 Altibase 서버 프로세스가 OOM Killer에 의해 비정상 종료할 가능성이 커진다.
- 권고값 1은 swappiness를 비활성화하지 않고 스와핑을 최소화하기 위한 설정이다.
- 페이지 캐시는 파일 I/O 성능 향상을 위해 Linux에서 관리하는 메모리 영역이다.

#### Linux 커널 버전

- swappiness는 커널 버전 상관없이 모두 적용한다.

## THP(Transparent Huge Pages)

THP는 대량의 메모리 관리를 위해 Linux에서 채택한 방법으로 4096 바이트 단위의 메모리 페이지를 2MB 또는 1GB 단위로 확대하는 기능을 자동화하는 설정이다.

하지만 Altibase 운용 환경에서 메모리 할당 지연 및 단편화로 인한 성능 이슈 사례가 보고되어 비활성화를 권고한다.

커널 파라미터	설명	권고값
THP (transparent_hugepage)	커널에서 관리하는 메모리 페이지 단위를 기존 4K에서 2M 또는 1G단위로 확대하는 기능을 자동화한다. 기본값은 always이다.	never

### Linux 커널 버전

- THP(Transparent Huge Pages)는 Red Hat Enterprise Linux 6 부터 지원하는 기능이다.
- RHEL 6 의 커널 버전은 2.6.32-71 이다.
- [Red Hat Enterprise Linux Release Dates](#)

## max\_map\_count

테라바이트 단위의 메모리 테이블을 운영할 경우 max\_map\_count 파라미터 제한으로 메모리 할당 실패할 수 있다. 이는 트랜잭션 실패 등 Altibase 운용에 심각한 영향을 줄 수 있으므로 충분히 큰 값으로 설정하는 것을 권고한다.

커널 파라미터	설명	권고값
max_map_count	프로세스가 사용할 수 있는 메모리 맵 영역의 최대 수를 의미한다. 대부분의 경우 기본값으로 65530 이 적절하나 애플리케이션에 이 파일 보다 많은 수를 매핑해야 하는 경우 이 값을 늘려야한다.	2147483647

이 파라미터 제한으로 메모리 할당이 실패할 경우,

Altibase 서버 트레이스 로그 altibase\_boot.log에 아래와 같은 메시지가 남을 수 있다.

Failed to mmap log file ( errno=ENOMEM(12), Not enough memory

어플리케이션에서는 Memory [iduMemMgr::malloc] failed. 와 같은 에러가 발생할 수 있다.

## 공유 메모리(Shared memory)

한 서버에서 여러 응용 프로그램이 서로 정보를 주고받아야 하는 경우 필요한 커널 파라미터이다.

- Altibase 서버와 클라이언트의 통신 방식이 IPC 또는 IPCDA 타입인 경우
- 두 개 이상의 Altibase 응용 프로그램이 IPC로 통신하는 경우

OS는 이를 위해 IPC(Inter Process Communication)라고 부르는 자원을 제공한다. 여러 IPC 자원 중 프로세스 간 정보를 주고 받기 위한 메모리 영역을 공유 메모리(Shared memory)라고 부른다. 공유 메모리는 사용자가 단위를 지정하여 한 개 이상의 영역으로 나누어 설정할 수 있으며 이를 세그먼트(segment)라고 부른다.

예를 들어 사용자는 10MB로 1 개의 세그먼트를 갖는 공유 메모리를 설정할 수도 있고 10MB로 10 개의 세그먼트로 구성하여 100MB의 공유 메모리를 설정할 수도 있다. 따라서, 세그먼트의 최대 크기나 개수와 같은 세그먼트 관련한 커널 파라미터 설정이 필요하다.

Linux에서 제공하는 공유 메모리 관련 파라미터와 Altibase에서 권장하는 값은 아래와 같다.

커널 파라미터	설명	권고값
shmmni	생성할 수 있는 공유 메모리 세그먼트의 최대 개수로 기본값은 4096 이다.	4096
shmmax	공유 메모리 세그먼트 한 개의 최대 크기로 단위는 바이트이다. x86 시스템의 경우 최소 설정값은 268435456 바이트(256MB)이고 64 비트 시스템의 경우 2147483648 바이트(2GB)이다.	2147483648

## 세마포어(Semaphore)

Altibase 서버와 클라이언트의 통신 방식이 IPC 또는 IPCDA 타입인 경우 프로세스 간 동기화 구현하기 위해 필요한 커널 파라미터이다.

세마포어는 IPC에서 공유 자원에 대한 접근을 제한하기 위해 OS에서 제공하는 자원이다. IPC 또는 IPCDA 통신 방식에서 공유 메모리를 통신 버퍼로 사용하기 때문에 이 자원에 대한 읽기/쓰기의 동시성 제어를 위해 세마포어 오퍼레이션을 사용한다. 세마포어 오퍼레이션에 따라 프로세스는 대기 또는 진행 상태가 될 수 있다. 세마포어 오퍼레이션은 동시다발적으로 발생하기 때문에 세마포어 개수 및 오퍼레이션에 대한 적합한 커널 파라미터 설정이 필요하다.

Linux에서 제공하는 세마포어 관련 파라미터와 Altibase에서 권장하는 값은 아래와 같다.

커널 파라미터	설명	권고값
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
semmns	운영체제에서 세마포어의 최대 개수이며 한 개에 16 바이트의 커널 메모리가 할당된다.	32000
semopm	semop 시스템 콜에서 처리하는 오퍼레이션의 최대 개수이다.	512



semmni	세마포어 세트의 최대 개수로 65535 이내에 설정 가능하며 한 세트당 84 바이트의 커널 메모리가 할당된다.	5029
--------	---	------

---

## 커널 파라미터 변경 방법

각 커널 파라미터의 설정값을 확인하는 방법과 변경 방법을 알아본다.

---

### CPU frequency Governor

Altibase에서 권고하는 CPU frequency Governor 설정은 performance이다.

#### 설정값 확인

RHEL 6 과 7 에서 공통으로 사용하는 확인 방법이다. 일반적으로 cat 명령어로 확인하며 cpupowerutils 패키지가 설치되어 있다면 cpupower 명령어로도 확인할 수 있다.

- cat 명령어

RHEL 6 과 7 에서 공통으로 사용하는 확인 방법이다.

```
$ cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor |  
sort -u  
performance
```

CPUfreq 드라이버가 설치되어 있지 않은 경우 아래와 같은 출력 결과를 보일 수 있다. 이 경우 CPU frequency Governor 설정은 고려하지 않아도 된다.

```
$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor  
cat: /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor: 그런  
파일이나 디렉터리가 없습니다
```

#### 참고 - CPUfreq governor 드라이버 설치 여부 확인 방법

##### ➤ Red Hat Enterprise Linux 6

```
$ lsmod | grep cpufreq  
acpi_cpufreq          7699  0  
freq_table            4936  1 acpi_cpufreq  
mperf                 1557  1 acpi_cpufreq
```

또는

```
$ cpupower frequency-info | grep driver  
driver: acpi-cpufreq
```

CPUfreq 드라이버가 설치되지 않은 경우 출력 결과는 아래와 같다.  
lsmod 출력 결과는 나오지 않는다.

```
$ lsmod | grep cpufreq  
$ cpupower frequency-info | grep driver  
no or unknown cpufreq driver is active on this CPU
```

##### ➤ Red Hat Enterprise Linux 7

```
$ cpupower frequency-info | grep driver  
driver: intel_pstate
```

- **cpupower 명령어**

cpupower 명령어 버전에 따라 출력 결과는 다를 수 있다.

```
$ cpupower frequency-info -policy
analyzing CPU 0:
  current policy: frequency should be within 1.20 GHz and 3.60 GHz.
                  The governor "performance" may decide which speed to
use
                  within this range.
```

CPUfreq 드라이버가 설치되어 있지 않은 경우 출력 결과는 아래와 같다.

```
$ cpupower frequency-info -policy
analyzing CPU 0:
  Unable to determine current policy
```

- **CPU 코어 별 클럭 속도 확인**

아래 명령어로 모든 CPU 코어의 클럭 속도가 일정하게 설정되었는지 확인한다.

```
$ grep MHz /proc/cpuinfo | sort -u
```

## 설정값 변경 방법

### Red Hat Enterprise Linux 6

Red Hat Enterprise Linux6 에서 CPU frequency governor를 변경하는 방법이다.

#### 1) 즉시 변경

governor 지시문과 cpupower 명령어를 이용한 설정은 온라인 상태에서 변경할 수 있으나 OS를 재시작하면 초기화된다.

- **cat 명령어를 이용한 governor 지시문**

아래 명령어를 CPU 코어 수만큼 반복 수행한다.

```
$ echo performance >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
$ echo performance >
/sys/devices/system/cpu/cpu1/cpufreq/scaling_governor

코어 수 만큼 반복 수행
```

- **cpupower 명령어**

```
$ cpupower frequency-set -g performance
```

- **tuned**

tuned 시스템 도구가 활성화된 시스템의 경우

[RHEL 7 CPUfreq Governor 영구 적용 tuned](#) 부분을 참고한다.

#### 참고 - tuned 시스템 도구 활성화 확인 방법

```
# tuned 서비스가 활성화 상태의 경우 출력 예시
$ tuned-adm active
Current active profile: throughput-performance
```

```
# tuned 서비스가 비활성화 상태의 경우 출력 예시지
$ tuned-adm active
Current active profile: default
Service tuned: disabled, stopped
Service ktune: disabled, stopped
```

## 2) 영구 적용

- tuned

[RHEL 7 CPUfreq Governor 영구 적용 tuned](#) 부분을 참고한다.

- rc.local

tuned 서비스가 비활성화인 경우 사용한다.

/etc/rc.d/rc.local 파일에 CPU 코어 수만큼 아래 명령어를 추가한다.

```
$ echo performance >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
$ echo performance >
/sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
# 코어 수 만큼 반복 수행
```

## Red Hat Enterprise Linux 7

Red Hat Enterprise Linux7 에서 CPU frequency governor를 변경하는 방법이다.

### 1) 즉시 변경

- cpupower 명령어

cpupower 명령어를 이용한 설정 변경은 온라인 상태에서 변경할 수 있으나 OS를 재시작하면 초기화된다.

```
$ cpupower frequency-set -g performance
```

### 2) 영구 적용

- tuned

Linux에서 기본 제공하는 throughput-performance나 latency-performance프로파일을 사용한다. 이 두 프로파일을 사용하면 아래와 같은 효과를 볼 수 있다.

- CPUfreq governor가 performance로 설정된다.
- CPU 클럭 속도가 최대로 고정된다.

참고로, Red Hat Enterprise Linux 7 의 기본 프로파일은 throughput-performance 이다.

throughput-performance나 latency-performance 프로파일로 설정하는 방법은 두 가지이다.

- ✓ active 프로파일에 throughput-performance profile을 include한다.

```
# 1. 활성화된 tuned profile을 확인한다.
# 다음은 tuned active profile이 throughput-
performance나 latency-performance가 아닌 경우의
예이다.
# 명령어 수행 예시이며 아래 결과에서 active profile은
virtual-guest이다.

$ tuned-adm active
Current active profile: virtual-guest

# 2. 프로파일 경로로 이동한다.
# tuned profile의 기본경로는
/usr/lib/tuned/profile name 이다.
# [main] 타입에 아래와 같이 include를 추가한다.
# [cpu] 타입이 있다면 삭제한다.

$ cd /usr/lib/tuned/virtual-guest
$ vi tuned.conf

[main]
include=throughput-performance

# 3. 변경한 프로파일을 적용한다.
$ tuned-adm profile virtual-guest
```

- ✓ active 프로파일을 throughput-performance로 변경한다.

```
$ tuned-adm profile throughput-performance
```

---

## RemoveIPC

RemoveIPC는 Red Hat Enterprise Linux 7.2 이상 버전에서 적용하며 권고값은 no이다.

### 설정값 확인

logind.conf 파일에서 RemoveIPC=no 로 설정되어 있는 지 확인한다.

```
$ grep RemoveIPC /etc/systemd/logind.conf
RemoveIPC=no
```

### 설정값 변경 방법

#### 1) 즉시 변경

RemoveIPC는 Linux에서 즉시 변경하는 방법을 제공하지 않는다.

#### 2) 영구 적용

/etc/systemd/logind.conf 파일을 편집기로 열고 RemoveIPC=no 로 변경한다.

```
$ vi /etc/systemd/logind.conf

# This file is part of systemd.
...
[Login]
...
RemoveIPC=no
```

OS를 재시작하거나 다음 명령을 수행하여 변경 사항을 적용한다.

```
$ systemctl restart systemd-logind.service
```

---

## swappiness

현재 설정값을 확인하여 권고설정이 안되어 있는 경우 아래 가이드를 참고하여 변경한다.

### 설정값 확인

설정값 확인은 Linux 버전 상관없이 동일하다. 다음 두 가지 방법으로 확인할 수 있다. Altibase의 권고 설정값은 1 이다.

- cat 명령어

```
$ cat /proc/sys/vm/swappiness
```

- sysctl 명령어

```
$ sysctl -a | grep swappiness
```

---

### 설정값 변경 방법

#### Red Hat Enterprise Linux 6

Red Hat Enterprise Linux6 에서 swappiness를 변경하는 방법이다.

#### 1) 즉시 변경

echo 또는 sysctl 명령어를 이용한다.

- echo 명령어

```
$ echo 1 > /proc/sys/vm/swappiness
```

- sysctl 명령어

```
$ sysctl -w vm.swappiness=1
```

#### 2) 영구 적용

sysctl.conf, rc.local 두 가지 방법으로 설정할 수 있으므로 모두 동일하게 적용한다.

- sysctl.conf

```
# sysctl.conf 파일에 vm.swappiness = 1 를 추가하거나 값을 변경한다.
$ vi /etc/sysctl.conf
vm.swappiness = 1
```

- rc.local

```
# rc.local 파일에 이 명령어를 추가한다.
$ vi /etc/rc.d/rc.local
echo 1 > /proc/sys/vm/swappiness
```

## Red Hat Enterprise Linux 7

Red Hat Enterprise Linux7 에서 swappiness를 변경하는 방법이다.

### 1) 즉시 변경

echo 또는 sysctl 명령어를 이용한다.

- echo 명령어

```
$ echo 1 > /proc/sys/vm/swappiness
```

- sysctl 명령어

```
$ sysctl -w vm.swappiness=1
```

### 2) 영구 적용

tuned profile과 sysctl.conf 두 가지 방법으로 설정할 수 있으므로 모두 동일하게 적용한다.

- tuned

```
# 활성화된 Tuned profile을 확인한다.
$ tuned-adm active

# 수행 예.
# 아래 결과에서 active profile은 throughput-performance이다.
$ tuned-adm active
Current active profile: throughput-performance

# 프로파일 경로로 이동한다.
# tuned profile의 기본경로는 /usr/lib/tuned/profile_name 이다.
# active profile이 throughput-performance 인 경우의 예.
$ cd /usr/lib/tuned/throughput-performance

# tuned.conf 파일을 연다.
#[sysctl] 타입에 swappiness 설정을 추가하거나 변경한다.
[sysctl]타입이 없으면 추가한다.

$ vi tuned.conf
[sysctl]
```

```
vm.swappiness=1

# 변경한 프로파일을 적용한다
$ tuned-adm profile 'profile_name'
```

- /etc/sysctl.conf

```
$ vi /etc/sysctl.conf
vm.swappiness = 1
```

변경사항을 확인하기 위해 OS를 재시작한다. OS 재시작은 모든 커널 파라미터 변경 후 한번에 수행해도 된다.

---

## THP

현재 설정값을 확인하여 권고설정이 안되어 있는 경우 아래 가이드를 참고하여 변경한다.

### 설정값 확인

설정값 확인은 Linux 버전 상관없이 동일하다. THP 설정값은 다음 세 가지 모두 확인해야 한다.

1) 권고설정 : *never*

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

2) *Hugepagesize*를 제외한 모든 값이 0 이어야 한다.

```
$ grep -i huge /proc/meminfo
AnonHugePages:      0 kB
HugePages_Total:    0
HugePages_Free:     0
HugePages_Rsvd:     0
HugePages_Surp:     0
Hugepagesize:       2048 kB
```

3) 출력 결과에 *transparent\_hugepage=never* 설정이 포함되어 있는지 확인한다.

```
$ cat /proc/cmdline
BOOT_IMAGE=vmlinuz-3.10.0-862.el7.x86_64 root=/dev/mapper/centos-root
ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap
rhgb quiet transparent_hugepage=never
```



### Red Hat Enterprise Linux 6

Red Hat Enterprise Linux6 에서 THP를 변경하는 방법이다.

#### 1) 즉시 변경

아래 명령어는 새로운 THP 생성 및 사용만 중지한다. 이전에 생성된 THP가 해제되지는 않으므로 완전히 없애기 위해서는 THP를 비활성화한 상태에서 OS 재시작을 해야한다.

```
$ echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
$ echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
```

#### 2) 영구 적용

grub.conf 파일과 rc.local 파일에 모두 동일하게 적용한다.

- /etc/grub.conf

grub.conf 파일에 transparent\_hugepage=never 설정을 추가한다.

```
$ vi /etc/grub.conf
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux 6 (2.6.32-504.el6.x86_64)
    root (hd0,0)
    # 아래 kernel 마지막 부분에 transparent_hugepage=never를 추가한다.
    kernel /vmlinuz-2.6.32-504.el6.x86_64 ro root=/dev/mapper/vg_os-
lv_os rd_NO_LUKS LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16
crashkernel=auto rd_LVM_LV=vg_os/lv_os KEYBOARDTYPE=pc KEYTABLE=us
rd_NO_DM rhgb quiet transparent_hugepage=never
    initrd /initramfs-2.6.32-504.el6.x86_64.img
```

- /etc/rc.d/rc.local

rc.local 파일에 아래와 같이 추가한다.

```
$ vi /etc/rc.d/rc.local
...
# 아래의 스크립트를 추가한다.
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

OS를 재시작하여 설정 상태를 확인한다. OS 재시작은 모든 커널 파라미터 변경 후 한번에 수행해도 된다.

## Red Hat Enterprise Linux 7

Red Hat Enterprise Linux7 에서 THP를 변경하는 방법이다.

### 1) 즉시 변경

RHEL 7 에서는 OS 구동 상태에서 THP 설정을 변경할 수 없다.

### 2) 영구 적용

RHEL 7 의 경우 grub, tuned 설정으로 활성화할 수 있으므로 모든 설정에서 비활성화한다.

grub, tuned 설정 후에는 변경사항을 확인하기 위해 OS를 재시작한다. OS 재시작은 모든 커널 파라미터 변경 후 한번에 수행해도 된다.

- grub

아래 순서대로 수행한다.

- i. /etc/default/grub 파일에 GRUB\_CMDLINE\_LINUX 항목에 transparent\_hugepage=never를 추가한다.

```
$ vi /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
# GRUB_CMDLINE_LINUX 항목에 마지막에 transparent_hugepage=never 를 추가
GRUB_CMDLINE_LINUX="nomodeset crashkernel=auto
rd.lvm.lv=vg_os/lv_root rd.lvm.lv=vg_os/lv_swap rhgb quiet
transparent_hugepage=never"
GRUB_DISABLE_RECOVERY="true"
```

- ii. grub2-mkconfig를 수행하여 grub.cfg 파일을 생성한다.

펌웨어 타입에 따라 grub.cfg 파일 생성 위치가 다르므로 다음 명령어를 참고하여 수행한다.

#### On BIOS-based machines

```
$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

#### On UEFI-based machines

```
$ grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

#### 참고 - UEFI 모드 확인 방법

```
# 1. grub.conf 의 위치
$ ls -la /etc/grub.conf
lrwxrwxrwx. 1 root root 32 2016-08-22 19:55 /etc/grub.conf -
> ../boot/efi/EFI/redhat/grub.conf

# 2. efi 디렉토리 존재 유무
$ ll /sys/firmware | grep ^d | grep -i efi
drwxr-xr-x 3 root root 0 2015-09-09 13:27 efi

# 3. Dmesg
$ dmesg |grep EFI
EFI v2.40 by EDK II
```

```
EFI: mem00: type=3, attr=0xf, range=[0x0000000000000000-0x0000000000001000) (0MB)
EFI: mem01: type=7, attr=0xf, range=[0x0000000000001000-0x0000000000001e000) (0MB)
EFI: mem02: type=2, attr=0xf, range=[0x0000000000001e000-0x00000000000020000) (0MB)
EFI: mem03: type=3, attr=0xf, range=[0x00000000000020000-0x0000000000009f000) (0MB)
...중략...
```

- **tuned**

tuned 서비스가 활성화되어 있다면 tuned profile에도 THP 설정을 비활성화해야 한다.

Red Hat Enterprise Linux 7 의 기본 tuned profile은 throughput-performance이고 THP 활성화가 기본 설정이다.

아래는 active tuned profile에서 THP를 비활성화하는 방법이다.

```
# 활성화된 tuned profile을 확인한다.
# 아래는 수행 예이며 아래 결과에서 active profile은 throughput-performance이다.
$ tuned-adm active
Current active profile: throughput-performance

# 프로파일 경로로 이동한다.
# tuned profile의 기본경로는 /usr/lib/tuned/profile_name 이다.
# active profile이 throughput-performance 인 경우의 예.
$ cd /usr/lib/tuned/throughput-performance

# tuned.conf 파일을 연다.
# [vm] 타입에 transparent_hugepage 설정을 추가한다. [vm] 타입이 없으면 추가한다.
$ vi tuned.conf
[vm]
transparent_hugepage=never

# 변경한 프로파일을 적용한다
$ tuned-adm profile 'profile_name'
```

새로운 tuned profile을 등록하는 방법도 있다.

<https://www.thegeekdiary.com/centos-rhel-7-how-to-disable-transparent-huge-pages-thp/>

## max\_map\_count

현재 설정값을 확인하여 권고설정이 안되어 있는 경우 아래 가이드를 참고하여 변경한다.

### 설정값 확인

설정값 확인은 Linux 버전 상관없이 동일하다. 다음 두 가지 방법으로 확인할 수 있다.

Altibase의 권고 설정값은 2147483647 이다.

- **cat 명령어**

```
$ cat /proc/sys/vm/max_map_count
```

- sysctl 명령어

```
$ sysctl -a | grep max_map_count
```

## 설정값 변경 방법

---

### Red Hat Enterprise Linux 6

Red Hat Enterprise Linux6 에서 max\_map\_count를 변경하는 방법이다.

#### 1) 즉시 변경

- sysctl 명령어를 이용한다.

```
$ sysctl -w vm.max_map_count=2147483647
```

#### 2) 영구 적용

sysctl.conf, rc.local 두 가지 방법으로 설정할 수 있으므로 모두 동일하게 적용한다.

- sysctl.conf

sysctl.conf 파일에 vm.max\_map\_count = 2147483647 를 추가하거나 값을 변경한다.

```
$ sysctl -w vm.max_map_count=2147483647
```

- rc.local

rc.local 파일에 이 명령어를 추가한다.

```
$ vi /etc/rc.d/rc.local  
echo 2147483647 > /proc/sys/vm/max_map_count
```

### Red Hat Enterprise Linux 7

Red Hat Enterprise Linux7 에서 max\_map\_count를 변경하는 방법이다.

#### 1) 즉시 변경

- sysctl 명령어를 이용한다.

```
$ sysctl -w vm.max_map_count=2147483647
```

#### 2) 영구 적용

tuned profile과 sysctl.conf 두 가지 방법으로 설정할 수 있으므로 모두 동일하게 적용한다.

grub, tuned 설정 후에는 변경사항을 확인하기 위해 OS를 재시작한다. OS 재시작은 모든 커널 파라미터 변경 후 한번에 수행해도 된다.

- tuned

```
# 활성화된 Tuned profile을 확인한다.
$ tuned-adm active

# 수행 예.
# 아래 결과에서 active profile은 throughput-performance이다.

$ tuned-adm active
Current active profile: throughput-performance

# 프로파일 경로로 이동한다.
# tuned profile의 기본경로는 /usr/lib/tuned/profile_name 이다.
# active profile이 throughput-performance 인 경우의 예.
$ cd /usr/lib/tuned/throughput-performance

# tuned.conf 파일을 연다.
# [sysctl] 타입에 swappiness설정을 추가하거나 변경한다.[sysctl] 타입이 없으면
# 추가한다.
$ vi tuned.conf
[sysctl]
vm.max_map_count=2147483647

# 변경한 프로파일을 적용한다
$ tuned-adm profile 'profile_name'
```

- /etc/sysctl.conf

```
$ vi /etc/sysctl.conf
vm.max_map_count=2147483647
```

## 공유 메모리와 세마포어

### 설정값 확인

공유 메모리와 세마포어의 설정값은 다음 두 가지 명령어로 확인할 수 있다.

- ipcs 명령어

```
# 공유 메모리 커널 파라미터 설정값 확인

$ ipcs -m -l
----- Shared Memory Limits -----
max number of segments = 4096           # shmmni (권고값 4096)
max seg size (kbytes) = 2097152         # shmmax (권고값 2097152)
max total shared memory (kbytes) = 137438953472
min seg size (bytes) = 1

# 세마포어 커널 파라미터 설정값 확인

$ ipcs -s -l
----- Semaphore Limits -----
max number of arrays = 5029             # semmni (권고값 5029)
max semaphores per array = 2000         # semmsl (권고값 2000)
```

```
max semaphores system wide = 32000      # semmns (권고값 32000)
max ops per semop call = 512            # semopm (권고값 512)
semaphore max value = 32767
```

- sysctl 명령어

```
# 명령문 및 수행 결과 예시

$ sysctl -a | grep -e kernel.shmmax -e kernel.shmmni -e kernel.sem
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 2000      32000   512   5029
```

## 설정값 변경 방법

### 1) 즉시 변경

- echo 명령어를 이용한다.

```
# 공유 메모리 설정

echo 4096 > /proc/sys/kernel/shmmni
echo 2147483648 > /proc/sys/kernel/shmmax

# 세마포어 설정

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

### 2) 영구 적용

- /etc/sysctl.conf 파일에 추가한다.

```
$ vi /etc/sysctl.conf
kernel.shmmni = 4096
kernel.shmmax = 2147483648
kernel.sem = 2000      32000   512   5029
```

변경사항을 확인하기 위해 OS를 재시작한다. OS 재시작은 모든 커널 파라미터 변경 후 한번에 수행해도 된다.

## 사용자 환경

Linux에서 Altibase 서버를 설치하고 운영하기 위해 필요한 사용자 환경 설정에 관한 설명이다. 사용자는 OS 사용자를 의미하며 Altibase 서버를 설치하고 Altibase 서버 프로세스를 구동하는 사용자이다.

배시 셸(Bash Shell)에서 사용자 로그인 환경설정 파일은 `.bash_profile`이다. 사용자 환경설정 파일은 셸에 따라 다르나 Linux 기본 셸이 배시 셸이므로 이를 기준으로 설명한다.

사용자 환경은 다음 두 가지로 나뉜다.

- 리소스 제한
- 환경 변수

## 리소스 제한

Linux는 CPU, 메모리, 파일 등과 같은 시스템 자원(리소스, resource)을 제한하는 설정을 제공한다. 특정 사용자가 시스템 자원을 독점하는 것을 막기 위한 목적이나 파일 생성 제한, 메모리 할당 제한 등으로 인해 Altibase 서버 운영에 치명적인 영향을 미치기 때문에 적절한 설정이 필요하다.

Altibase 서버 운영 환경에서 권고하는 리소스 항목과 권고값은 아래와 같다.

리소스 종류		설명	권장값
ulimit 명령어	limits.conf item		
data seg size (kbytes, -d)	data	프로세스 데이터 영역의 최대 메모리 크기	unlimited
file size (blocks, -f)	fsiz	생성 가능한 파일의 최대 크기	unlimited
open files (-n)	nofile	한 프로세스에서 열 수 있는 최대 open file descriptor 수	1048576
max memory size (kbytes, -m)	rss	사용 가능한 메모리의 최대 크기	unlimited
virtual memory (kbytes, -v)	as	사용 가능한 가상 메모리의 최대 크기	unlimited
max user processes (-u)	nproc	사용자가 실행할 수 있는 프로세스(쓰레드 포함)의 최대 개수	unlimited

### 리소스 관련 에러 메시지

리소스 제한으로 발생하는 에러 메시지이다. Altibase 서버 운영 중 아래의 에러 메시지가 확인되면 리소스 제한 설정을 확인할 필요가 있다.

- max user processes 가 부족한 경우
  - Failed to create a thread object.

- resource temporarily unavailable
- open files 가 부족한 경우
  - Too many open files

## 리소스 설정값 확인 및 변경

리소스 제한은 사용자 별로 설정하지만, 사용자 프로세스에 개별 적용된다.

변경한 리소스 제한 설정값을 Altibase 서버에 반영하려면 Altibase 서버를 재기동해야 한다.

### 설정값 확인 방법

- 리소스 제한 설정값을 확인하는 명령어이다.

```
# -S 또는 -H 옵션을 명시하지 않으면 -S(Soft-Limit)을 출력한다.
$ ulimit -a

# Hard-Limit 설정값을 출력한다.
$ ulimit -Ha
```

- Altibase 서버 프로세스에 설정된 리소스 제한값은 아래 명령어로 확인한다.

```
$ cat /proc/`ps -ef | grep 'altibase -p' | grep -v grep | awk '{print $2}'`/limits
```

- Altibase 서버 프로세스에 원하는 설정값이 적용되어 있지 않다면 Altibase 서버를 재기동해야 한다.

### 설정값 변경 방법

1. **ulimit 명령을 수행한다.**

ulimit은 리소스 제한을 설정하는 명령어이다. Altibase 서버를 설치하고 구동하는 OS 사용자의 환경설정 파일 .bash\_profile에 ulimit 설정 명령어를 추가한다.

사용자 리소스 제한 설정 예제 - .bash\_profile 에 추가

```
$ vi ~/.bash_profile #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
```

2. **환경설정 파일(.bash\_profile)을 적용한다.**

아래 명령어로 ulimit 설정을 적용한다.

사용자 리소스 제한 설정 예제 - .bash\_profile 에 추가

```
$ . ~/.bash_profile
```



환경설정 파일 적용 시 아래와 같은 에러가 발생할 수 있다.

```
-bash: ulimit: open files: cannot modify limit: 명령을 허용하지 않음
-bash: ulimit: max user processes: cannot modify limit: 명령을 허용하지 않음
```

이는 Hard-Limit에 걸려 발생하는 에러이다.

### Hard-Limit & Soft-Limit

시스템 리소스 제한에는 Hard-Limit과 Soft-Limit이 있다. Hard-Limit은 OS 일반 사용자가 설정할 수 있는 최대 리소스 제한 설정값을 의미하고 Soft-Limit은 사용자에게 설정된 리소스 제한값을 의미한다.

Hard-Limit이 사용자가 ulimit으로 설정하려는 값보다 작으면 cannot modify limit 에러가 발생한다.

#### Hard-Limit 확인 방법

```
$ ulimit -H -u -n    # -H 옵션이 Hard-Limit을 의미한다.
max user processes   (-u) 1024
open files            (-n) 4096
```

#### /etc/security/limits.conf 변경

Hard-Limit 변경은 root 권한이 필요하다. limits.conf 파일에 아래 설정을 추가한 후 저장한다.

#### Hard-Limit 설정 예제 - OS 사용자 이름이 altibase 인 경우

```
#<domain>      <type>  <item>      <value>
altibase        -      nofile      1048576
altibase        -      nproc       unlimited
```

해당 OS 유저로 로그인하여 2.단계를 다시 수행한다.

## 환경변수

Altibase 서버 설치 후 설정해야 할 환경변수이다.

구분	환경변수	설명	설정값
필수	자동 설정	ALTIBASE_HOME	Altibase를 설치한 경로를 지정한다.
		PATH	Altibase의 유틸리티 및 셸 스크립트의 위치를 찾기 위한 환경변수이다.
		LD_LIBRARY_PATH	Altibase 동적 라이브러리의 위치를 찾기 위한 환경변수이다.
		CLASSPATH	Java Class 파일의 위치를 찾기 위한 환경변수이다.
	수동 설정	ALTIBASE_NLS_USE	Altibase 클라이언트 캐릭터셋 설정을 위한 Altibase 서버 캐릭터셋과 동일

			환경변수이다.  Altibase 서버 캐릭터셋과 동일하게 설정한다.	
		LANG	사용자의 시스템 로케일을 정의한다.	Altibase 서버 캐릭터셋에 영향을 받는다.
선택	수동 설정	MALLOC_ARENA_MAX	<ul style="list-style-type: none"> <li>멀티쓰레드 응용 프로그램 환경에서 스레드 간 메모리 경합으로 인한 성능 이슈를 개선하기 위해 Red Hat Enterprise Linux 6 에서 추가된 기능이다.</li> <li>기본값은 CPU 코어 수 * MALLOC_ARENA_TEST 이다.</li> <li>MALLOC_ARENA_TEST 환경변수 기본값 <ul style="list-style-type: none"> <li>32-bit 의 경우 2</li> <li>64-bit 의 경우 8</li> </ul> </li> <li>MALLOC_ARENA_MAX 환경변수는 glibc2.10 이상에서 올바르게 동작한다.</li> </ul>	-

- 환경변수 MALLOC\_ARENA\_MAX
  - 이 환경변수는 선택사항이다.
  - 일반적으로 기본값을 유지하고 메모리 이슈가 발견될 경우 기본값보다 작게 설정하는 것을 고려한다.
  - 이 환경변수는 설정값에 따라 Altibase 서버 프로세스의 성능과 메모리 사용량에 영향을 미치기 때문에 동일한 값을 일괄적으로 권고하기 어렵다.
  - 이 환경변수의 기본값은 CPU 코어 수에 영향을 받아 CPU 코어 수가 많을수록 Altibase 서버 프로세스의 VSZ이 과도하게 커지는 현상이 발생할 수 있다.

#### 환경변수 설정값 확인 방법

필수 환경변수가 올바르게 설정되어 있는지 확인한다.

- env 명령어

env 결과는 해당 세션에 설정된 모든 환경변수를 출력한다.

```
$ env
```

- **echo 명령어**

지정한 환경변수 설정값을 출력한다. 값을 출력하지 않으면 환경변수가 설정되어 있지 않은 것을 의미한다.

```
# echo 명령어를 이용하여 ALTIBASE_HOME 환경변수 설정값을 확인하는 예이다.
$ echo $ALTIBASE_HOME
```

## 환경변수 설정값 변경 방법

필수 환경변수 중 수동으로 설정해야 하는 ALTIBASE\_NLS\_USE, LANG 환경변수를 설정하는 방법이다.

1. Altibase 서버 캐릭터셋을 확인한다.

```
iSQL> SELECT NLS_CHARACTERSET FROM V$NLS_PARAMETERS;
```

2. .bash\_profile 파일에 환경변수를 설정한다.

환경변수는 OS 사용자 세션 단위로 설정된다. 따라서 OS 사용자가 접속할 때마다 적용하기 위해 사용자 환경설정 파일 .bash\_profile에 추가해야 한다.

아래 표는 Altibase 서버 캐릭터셋에 따른 ALTIBASE\_NLS\_USE, LANG 환경변수 설정값을 보여준다.

Altibase 서버 캐릭터셋	ALTIBASE_NLS_USE	LANG
KO16KSC5601	KO16KSC5601	ko_KR.euckr
MS949	MS949	ko_KR.euckr
UTF8	UTF8	ko_KR.utf8

위 표를 참고하여 사용자 환경설정 파일 .bash\_profile에 환경변수를 추가한다.

```
# Altibase 서버 캐릭터셋이 UTF8 인 경우 예제이다.
# .bash_profile에 아래 내용을 추가하고 저장한다.

export ALTIBASE_NLS_USE=UTF8
export LANG=ko_KR.utf8
```

3. 환경설정 파일(.bash\_profile)을 적용한다.

아래 명령어로 추가한 환경변수를 적용한다.

```
$ . ~/.bash_profile
```

## 요약

앞에서 소개한 커널 파라미터와 환경변수의 권고값을 확인하는 방법을 표로 정리하였다.

분류		권고설정	확인방법
커널 파라미터	CPU frequency Governor	performance	cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor   sort -u
		CPU 코어의 클럭 속도가 최대로 고정	grep MHz /proc/cpuinfo   sort -u
	RemoveIPC	no	grep RemoveIPC /etc/systemd/logind.conf
	swappiness	1	cat /proc/sys/vm/swappiness 또는 sysctl -a   grep swappiness
	THP	never	cat /sys/kernel/mm/transparent_hugepage/enabled
		Hugepagesize 외에 모두 0	grep -i huge /proc/meminfo
		transparent_hugepage=never 포함	cat /proc/cmdline
	max_map_count	2147483647	cat /proc/sys/vm/max_map_count
공유 메모리	shmmni	4096	ipcs -m -l
	shmmax	2147483648	sysctl -a   grep -e kernel.shmmax -e kernel.shmmni
세마포어	semmsl	2000	ipcs -s -l sysctl -a   grep -kernel.sem
	semmns	32000	
	semopm	512	
	semmni	5029	
OS 사용자 리소스 제한	data seg size (kbytes, -d)	unlimited	ulimit -a
	file size (blocks, -f)	unlimited	
	open files (-n)	1048576	
	max memory size (Kbytes, -m)	unlimited	
	virtual memory (Kbytes, -v)	unlimited	
	max user processes (-u)	unlimited	
OS 사용자 환경변수 - 필수	ALTIBASE_HOME	Altibase 설치 경로 (절대 경로)	env
	PATH	\$ALTIBASE_HOME/bin	
	LD_LIBRARY_PATH	\$ALTIBASE_HOME/lib	
	CLASSPATH	\$ALTIBASE_HOME/lib	
	ALTIBASE_NLS_USE	Altibase 서버 캐릭터셋과 동일	
	LANG	Altibase 서버 캐릭터셋과 동일	

---

### Red Hat Enterprise Linux 권장 Swap 크기

Red Hat Linux에서는 Swap 크기 설정에 대해 아래와 같이 권고하고 있다.

- 과거에는 물리 메모리의 2 배의 Swap 공간을 권고했으나 테라바이트 급의 메모리를 가진 오늘날에는 과거 권고는 실용성이 없다.
- 140 개 이상의 논리 프로세서를 가진 시스템 혹은 3TB 이상의 RAM을 가진 시스템은 최소 100GB의 Swap 공간을 권장한다.

구체적인 내용은 Red Hat CUSTOMER PORTAL 페이지에서 참고한다.

- [Red Hat Enterprise Linux 권장 swap 크기는 어떻게 되나요?](#)

---

## 참고자료

---

### Linux 호환성

- [Altibase 7.1 Korean manuals - Installation Guide#OS-Patch](#)
- [Bug 1244002 - NFS and Fuse mounts hang while running IO - Malloc/free deadlock](#)
- [Altibase HDB 가 지원하는 플랫폼\(OS\)은?](#)

---

### CPU frequency Governor

- [Red Hat Enterprise Linux - 6 - 전원 관리 가이드 - 3.2. CPUfreq 조정기 사용하기](#)
- [Red Hat Enterprise Linux - 6 - 전원 관리 가이드 - 2.5.2. Tuned-adm](#)
- [Red Hat Enterprise Linux - 6 - 성능 조정 가이드 - 7.2. 파일 시스템 성능 프로파일](#)
- [https://access.redhat.com/documentation/ko-kr/red\\_hat\\_enterprise\\_linux/6/html-single/power\\_management\\_guide/index#cpufreq\\_setup](https://access.redhat.com/documentation/ko-kr/red_hat_enterprise_linux/6/html-single/power_management_guide/index#cpufreq_setup)
- [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html-single/power\\_management\\_guide/index#cpufreq\\_drivers](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/power_management_guide/index#cpufreq_drivers)
- [https://access.redhat.com/documentation/ko-kr/red\\_hat\\_enterprise\\_linux/6/html-single/power\\_management\\_guide/index](https://access.redhat.com/documentation/ko-kr/red_hat_enterprise_linux/6/html-single/power_management_guide/index)
- [https://access.redhat.com/documentation/ko-kr/red\\_hat\\_enterprise\\_linux/7/html-single/performance\\_tuning\\_guide/index](https://access.redhat.com/documentation/ko-kr/red_hat_enterprise_linux/7/html-single/performance_tuning_guide/index)
- [RHEL7: How to get started with CPU governor.](#)

---

### RemoveIPC

- [logind.conf\(5\) - Linux manual page - man7.org](#)
- [Daemons using IPC terminate unexpectedly after update to Red Hat Enterprise Linux 7.2](#)
- [Database Installation and Operation Fails if RemoveIPC=yes Is Configured for systemd-Oracle docs](#)
- [IBM® Db2® servers might crash when the RemoveIPC option for Red Hat Enterprise Linux Version 7.2 is set to yes.](#)

---

### swappiness

- [Wikipedia-Paging#Swappiness](#)
- [Wikipedia-Page\\_cache](#)
- [Wikipedia-Talk:Swappiness](#)
- [Red Hat Enterprise Linux 권장 swap 크기는 어떻게 되나요?](#)
- [Red Hat Enterprise Linux 7 - 성능 조정 가이드 - 4.3. 설정 도구](#)

---

### THP

- [Red Hat Enterprise Linux-Huge Pages 및 Transparent Huge Pages](#)
- [THP\(Transparent Huge Page\)때문에 System Hang](#)
- [How to use, monitor, and disable transparent hugepages in Red Hat Enterprise Linux 6 and 7?](#)
- [Disabling Transparent HugePages - Oracle Docs](#)
- [Transparent Huge Pages on Linux - SAP Help Portal](#)
- [Red Hat Enterprise Linux 6 で THP \(transparent hugepages\) を無効にしても反映されない](#)
- [Red Hat Enterprise Linux 7 で transparent hugepages \(THP\) を無効にする](#)
- [CentOS / RHEL 6 : How to disable Transparent Huge pages \(THP\)](#)
- [CentOS / RHEL 7 : How to disable Transparent Huge pages \(THP\)](#)

---

## max\_map\_count

- <https://www.kernel.org/doc/Documentation/sysctl/vm.txt>
- [Red Hat Enterprise Linux 7 - Performance Tuning Guide - 7.5. Configuring System Memory Capacity](#)
- [Additional OS validations required for SAP Applications on RHEL 7.++](#)

---

## 공유 메모리 및 세마포어

- IPCDA 는 Altibase 7 버전부터 지원하는 통신방식이다.
- [Altibase 7.1 Korean manuals - Administrator's Manual 2#12 서버클라이언트-통신](#)
- [Altibase 7.1 Korean manuals - Installation Guide#OS 별-커널-파라미터-설정](#)

---

## 로케일(Locale)

- <https://wiki.archlinux.org/index.php/Locale> (한국어)

---

## MALLOC\_ARENA\_MAX

- [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/6/html/6.0\\_release\\_notes/compiler](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/6.0_release_notes/compiler)
- [https://www.ibm.com/developerworks/community/blogs/kevgrig/entry/linux\\_glibc\\_2\\_10\\_rhel\\_6\\_malloc\\_may\\_show\\_excessive\\_virtual\\_memory\\_usage?lang=en](https://www.ibm.com/developerworks/community/blogs/kevgrig/entry/linux_glibc_2_10_rhel_6_malloc_may_show_excessive_virtual_memory_usage?lang=en)
- <https://github.com/prestodb/presto/issues/8993>



#### **알티베이스㈜**

서울특별시 구로구 디지털로 306

대륭포스트 2 차 10 층

02-2082-1000

<http://www.altibase.com>

#### **기술지원서비스**

02-2082-1114

<http://support.altibase.com/kr/>

Copyright © 2000~2010 ALTIBASE Corporation. All Rights Reserved.

이 문서는 정보 제공을 목적으로 제공되며, 사전에 예고 없이 변경될 수 있습니다. 이 문서는 오류가 있을 수 있으며, 상업적 또는 특정 목적에 부합하는 명시적, 묵시적인 책임이 일체 없습니다. 이 문서에 포함된 Altibase 제품의 특징이나 기능의 개발, 발표 등의 시기는 ㈜알티베이스 재량입니다. ㈜알티베이스는 이 문서에 대하여 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권을 보유할 수 있습니다.