



리눅스 시스템 성능 향상을 위한 커널 매개 변수 분석

배유미¹ · 박정수² · 신병웅³ · 정성재^{4*}

¹한남대학교 탈메지이교양교육대학

²코오롱베니트(주)

³Dell EMC systems

⁴(주)엔버 기업부설연구소

Analyze kernel parameters for improving Linux system performance

Yu-Mi Bae¹ · Jeong-Su Park² · Byung-Woong Shin³ · Sung-Jae Jung^{4*}

¹Talmage Liberal Arts College, Hannam University, Daejeon 34430, Korea

²Kolon Benit Co., Ltd, Seoul 13837, Korea

³Dell EMC systems, Seoul 06236, Korea

⁴Research Institute, Enber Co., Ltd, Seoul 04929, Korea

[요약]

리눅스 운영체제는 커널 매개 변수를 통해 시스템 관리자가 시스템의 상황에 맞게 다양한 설정을 할 수 있도록 제공하고 있다. 특히 파일 시스템 관련 매개 변수, 세마포 및 공유 메모리와 같은 IPC 관련 매개 변수, 네트워크 관련 매개 변수, 스왑 및 페이지 같은 메모리 관련 변수를 통해 시스템의 성능을 향상시키고 보안을 강화시킬 수 있다. 가장 최근에 등장한 엔터프라이즈 리눅스인 RHEL 7 버전에서는 지원하는 시스템의 성능이 향상되면서 관련 커널 매개 변수의 수치값도 크게 상승하고 있다. RHEL 7에서는 성능적 향상에 따라 지원되는 최대값을 증가시켰지만, 보안 강화를 위해 기본적으로 설정되는 값의 수치는 낮추었다. 본 논문에서는 RHEL 7 버전을 기반으로 시스템 성능 향상 및 보안과 관련 있는 다양한 커널 매개 변수를 RHEL 6 버전과 비교 분석한다.

[Abstract]

The Linux operating system provides system administrators with a variety of kernel parameters to make various configurations for the system. In particular, memory-related variables such as file system-related parameters, IPC-related parameters such as semaphores and shared memory, network-related parameters, swap and pages can improve system performance and enhance security. In the latest version of enterprise Linux, RHEL 7, the numeric value of the relevant kernel parameters is also increasing significantly as the performance of the supported system improves. In RHEL 7, the maximum value supported is increased with performance improvement, but the value set by default is lowered for security enhancement. In this paper, we compare RHEL 6 version with various kernel parameters related to system performance enhancement and security based on RHEL 7 version.

색인어 : 커널 매개 변수, 리눅스, 리눅스 성능, 레드햇 엔터프라이즈 리눅스 7, 시스템 분석

Key word : Kernel parameter, Linux, Linux performance, RHEL 7, System analysis

<http://dx.doi.org/10.9728/dcs.2018.19.10.1957>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 13 September 2018; **Revised** 10 October 2018

Accepted 22 October 2018

***Corresponding Author; Sung-Jae Jung**

Tel: +8

E-mail: posein@naver.com

I. 서론

공개형 운영체제인 리눅스는 소스가 공개되어있고 무료로 사용할 수 있다는 장점으로 인해 300개 이상의 다양한 배포판이 등장하게 되었다[1]. 이로 인해 사용 분야도 데스크톱 및 서버 분야뿐만 아니라 모바일 기기, TV 및 냉장고와 같은 가전 제품, 자동차, IoT(internet of things) 등 거의 모든 영역에서 중요한 위치를 차지하고 있다[2][3]. 특히 서버 분야를 비롯하여 클라우드 컴퓨팅 및 빅 데이터와 같이 IT 인프라 영역에서의 리눅스 점유율은 유닉스(unix) 운영체제가 차지했던 시장을 대체하면서 더욱 끌어올리는 추세이다. 이러한 추세는 엔터프라이즈 리눅스 서버 분야에서 독보적인 위치에 있는 레드햇(red hat) 사에 Red Hat Enterprise Linux(이하 RHEL)의 7 버전을 출시하면서 이러한 현상은 더욱 두드러지게 나타나고 있다. RHEL 7에서는 GUI(graphic user interface) 기반의 사용자 편의성 강화와 효율성 강화를 위해 전통적으로 사용되는 X Window 시스템 대신에 웨일랜드(wayland)를 도입하였다[4]. 또한 기본 파일 시스템을 EXT에서 XFS로 변경하면서 지원하는 파일 시스템의 크기, 파일 크기 등을 대폭 증가시켰다. 단순히 지원되는 사양만 증가한 것이 아니라 데몬(daemon) 기반으로 동작하면서 고용량 디스크 및 SSD(solid state drive)와 같은 고성능 디스크에서 탁월한 성능을 나타내고 있다[5]. 이러한 성능적인 향상은 리눅스의 커널 매개 변수(kernel parameter)에도 영향을 주고 있다. 리눅스 시스템은 핵심적인 역할을 수행하는 커널과 관련된 다양한 매개 변수를 제공하고 있다. 리눅스 관리자는 커널 매개 변수의 값을 변경함으로써 시스템의 성능을 향상시키고 보안을 강화할 수 있다. 본 논문에서는 레드햇의 상용 버전 리눅스인 RHEL 7을 기반으로 이전 버전인 RHEL 6과 비교하면서 시스템의 성능 향상과 관련된 리눅스 커널 매개 변수를 분석한다.

II. 관련 정보

2-1 커널 매개 변수의 개요

리눅스는 커널 제어를 위한 매개 변수들을 보관하고 관리하는 디렉터리는 /proc/sys 이다. 이 디렉터리 안에는 관련 항목별로 여러 개의 하위 디렉터리가 존재하는 데 주요 디렉터리는 그림 1과 같다. 이 중에서 시스템의 성능 및 보안과 관련된 디렉터리로는 kernel, fs, net, vm을 손꼽을 수 있다. fs는 커널에서 최대 사용 가능한 파일의 수 지정과 같은 파일 시스템 관련 매개 변수가 들어 있는 디렉터리이고, kernel은 IPC(inter-process communication)와 관련된 Internal Message Queue, Semaphore, Shared Memory 등을 설정하는 매개 변수가 들어 있다. net은 TCP, ICMP 등 네트워크 관련 매개 변수가 들어 있다. 마지막으로 vm은 스왑(swap), 페이지(page) 등 메모리 관련 변수가 들어 있는 디렉터리이다.



그림 1. /proc/sys의 하위 디렉터리
Fig. 1. Subdirectories of /proc/sys

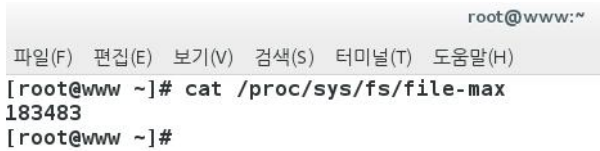


그림 2. 커널 매개 변수 값의 확인 예
Fig. 2. Example of checking kernel parameter values

2-2 커널 매개 변수 값의 확인 및 변경

커널 매개 변수 값을 확인하는 가장 쉬운 방법은 cat 명령을 사용하는 것이다. 예를 들어 커널에서 최대 사용 가능한 파일의 수를 확인하려면 그림 2와 같이 'cat /proc/sys/fs/file-max' 명령을 실행하면 된다[6]. 일반적으로 매개 변수에 해당하는 수치값이 명시되어 있는데, 매개 변수와 관련된 기능을 활성화(on) 또는 비활성화(off)시키는 경우에는 수치값으로 명시한다. 활성화인 경우에는 1(yes), 비활성화인 경우에는 0(no)으로 표기한다. 커널 매개 변수의 값을 변경하는 방법은 뒤에서 설명할 sysctl 명령어를 이용하는 방법도 있지만, 가장 쉬운 방법은 /proc/sys 디렉터리 하위에 있는 항목들을 echo 명령과 > 기호를 이용해서 변경하는 것이다. 예를 들어 ping 명령에 응답하지 않도록 설정하려면 관련 파일인 /proc/sys/net/ipv4/icmp_echo_ignore_all의 값을 변경하면 된다. 파일 이름을 해석해보면 icmp 패킷에 대한 응답을 전부 무시할 것이냐는 뜻이 되는데, 기본 설정값이 0(no)으로 ping에 응답을 하게 된다. 따라서 이 값을 1(yes)로 변경하면 응답을 하지 않게 된다.

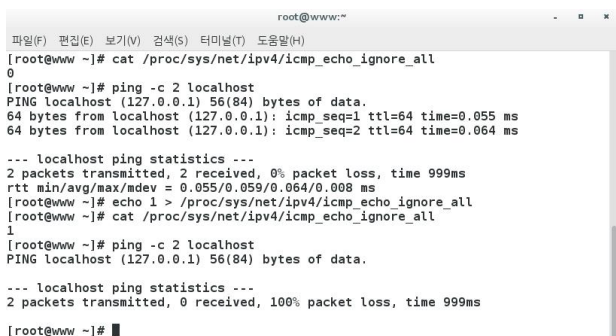


그림 3. 커널 매개 변수 값의 변경 예
Fig. 3. Example of changing kernel parameter values

2-3 IPC(inter-process communication)

시스템 성능 향상과 관련된 항목으로 IPC를 손꼽을 수 있다. IPC는 프로세스 간 통신이라는 뜻으로 프로세스들 사이에서 데이터를 주고받는 행위 또는 그에 대한 방법이나 경로를 통칭하는 말이다. IPC 방식에는 파일(file), 신호(signal), 소켓(socket), 공유 메모리(shared memory), 메시지 큐(message queue), 세마포(semaphore), 파이프(pipe) 등 다양하다[7].

1) 공유 메모리

공유 메모리 기법은 통신하는 프로세스들 사이에 특정 변수를 공유하고 이 공유 변수를 이용해서 정보를 교환하는 방법이다. 공유 메모리(shared memory)는 사용자가 단위를 지정하여 한 개 이상의 영역을 나누어 설정할 수 있다. 한 개의 공유 메모리 영역을 세그먼트(segment)라고 부른다. 예를 들어 사용자는 10MB로 1개의 세그먼트를 갖는 공유 메모리를 설정할 수 있고 10MB 영역을 10개의 세그먼트를 구성하여 100MB의 공유 메모리를 설정할 수 있다[8].

2) 메시지 시스템 기법

메시지 시스템 기법은 운영체제가 프로세스간의 메시지를 교환할 수 있도록 관리하는 방법으로 메시지를 메시지 큐에 저장하여 전달하는 방법이다.

3) 세마포(semaphore)

운영체제 상에서 프로세스들이 공유 자원에 대한 접근을 제어하기 위해 사용되는 신호를 의미한다. 세마포는 프로세스간의 동기화를 구현하기 위해 사용되는데, 동기화란 특정자원이나 객체에 대해서 한 시점에 하나의 프로세스만이 접근하여 변경 가능하도록 제어하는 것을 의미한다. 일반적으로 조회만 하는 경우에는 동시에 사용 가능하다. 다중 프로그래밍 환경에서 1개의 공유되는 자원에 제한된 개수의 프로세스 또는 스레드만 접근할 수 있도록 한다. 일반적으로 세마포의 카운트를 1이상으로 설정하고 카운트를 조절하여 진입 가능한 프로세스 및 스레드의 수를 조절한다.

4) 상호 배제(mutual exclusion, mutex)

여러 프로세스들이 하나의 공유 자원을 상호 배타적으로 사용할 수 있으나, 동시에 사용할 수 없도록 하는 것이다.

5) 관련 명령어: ipcs

리눅스 상에서 IPC 관련 항목인 메시지 큐, 공유 메모리 세그먼트, 세마포 관련 정보를 확인할 때 사용하는 명령이 ipcs이다. 그림 4와 같이 ‘ipcs -l’이라고 실행하면 관련 정보를 확인할 수 있고, 결과로 나오는 항목에 정보는 표 1과 같다.

```

posein@www:~$ ipcs -l
----- Messages Limits -----
max queues system wide = 3679
max size of message (bytes) = 8192
default max size of queue (bytes) = 16384

----- Shared Memory Limits -----
max number of segments = 4096
max seg size (kbytes) = 18014398509465599
max total shared memory (kbytes) = 18014398442373116
min seg size (bytes) = 1

----- Semaphore Limits -----
max number of arrays = 128
max semaphores per array = 250
max semaphores system wide = 32000
max ops per semop call = 32
semaphore max value = 32767

[posein@www ~]$
    
```

그림 4. ipcs 명령의 실행 결과

Fig. 4. Execution results of the ipcs command

표 1. ipcs 명령 결과의 항목

Table 1. Items in ipcs command result

item	comment
SHMMNI	max number of segments
SHMMAX	max seg size
SHMALL	max total shared memory
SHMMIN	min seg size
SEMMNI	max number of arrays
SEMMSL	max semaphores per array
SEMMNS	max semaphores system wide
SEMOPM	max ops per semop call
SEMVMX	semaphore max value

2-4 sysctl

sysctl 명령은 커널 매개 변수의 값을 제어하여 시스템을 최적화할 수 있는 명령이다. 즉 /proc/sys 디렉터리에 존재하는 매개 변수를 제어하는 명령이다. sysctl에서 매개 변수 호출하는 형식은 /proc/sys를 최상위 디렉터리(root)로 인식하고, 하위 디렉터리의 구분은 '.'으로 대체하여 명령을 수행한다. 예를 들면 커널 매개 변수 파일이 /proc/sys/net/ipv4/icmp_echo_ignore_all 이라면 net.ipv4.icmp_echo_ignore_all로 대체해서 사용한다. sysctl 명령으로 커널 매개 변수값을 확인하거나 변경 가능하지만 재부팅하면 초기화된다. 해당 설정을 재부팅 시에도 계속적으로 반영되게 하려면 /etc/sysctl.conf 파일에 등록해야 한다. /etc/sysctl.conf 파일은 vi 등 텍스트 편집기를 사용하면 되는데, #과 ;으로 시작하는 줄은 주석 처리되고, 기본적으로 ‘매개변수 = 값’ 형식으로 지정한다. 등록 후에 즉시 반영되게 하려면 ‘sysctl -p’ 명령을 실행하면 된다. 추가적으로 RHEL 6 이전 버전에서는 커널 매개변수 환경설정 파일로 /etc/sysctl.conf만을 사용하였다. 그러나 RHEL 7 버전에서는 /etc/sysctl.conf 이외에도 /usr/lib/sysctl.d/, /run/sysctl.d/, /etc/sysctl.d/, /usr/local/lib/sysctl.d/, /lib/sysctl.d/ 디렉터리 안의 conf 파일들도 사용한다. 따라서 /etc/sysctl.conf 파일은 주로 사용자가 설정하는 커널 매개 변수를 등록해서 사용하고 있다.

```

root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# sysctl -n net.ipv4.icmp_echo_ignore_all
0
[root@localhost ~]# sysctl -w net.ipv4.icmp_echo_ignore_all=1
net.ipv4.icmp_echo_ignore_all = 1
[root@localhost ~]# sysctl -n net.ipv4.icmp_echo_ignore_all
1
[root@localhost ~]#
    
```

그림 5. sysctl 명령의 사용 예
 Fig. 5. Example of the sysctl command

```

root@localhost:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# vi /etc/sysctl.conf
[root@localhost ~]# tail -1 /etc/sysctl.conf
net.ipv4.icmp_echo_ignore_all=1
[root@localhost ~]# sysctl -p
net.ipv4.icmp_echo_ignore_all = 1
[root@localhost ~]#
    
```

그림 6. /etc/sysctl.conf 파일의 사용 예
 Fig. 6. Example of the /etc/sysctl.conf file

III. 주요 커널 매개 변수 분석

리눅스 시스템의 성능 향상과 관련된 커널 매개 변수를 분류해보면 크게 파일 시스템, IPC, 네트워크와 같이 3가지 영역으로 나눌 수 있다.

3-1 파일 시스템 관련 매개 변수

파일 시스템 관련 커널 매개 변수는 /proc/sys/fs 디렉터리에 들어 있는데, 시스템 성능과 연관된 변수로 file-max와 file-nr이 있다. file-max는 커널에서 최대 사용 가능한 파일의 숫자값이 들어 있고, file-nr은 파일을 열 때 사용하는 파일 디스크립터(file descriptor) 관련 값이 들어 있다. file-nr은 표 2와 같이 총 3개의 값이 들어 있다. 첫 번째 값은 할당된 파일 디스크립터 수, 두 번째 값은 할당되지 않은 파일 디스크립터 수로 커널 2.6 버전부터는 항상 0으로 표기되어 있다. 마지막 값이 최대로 열 수 있는 파일 디스크립터 수로 file-max값에 해당한다.

표 2. 파일 시스템 관련 커널 매개 변수
 Table 2. File system related kernel parameters

parameter	default value
file-max	183483
file-nr	6592 0 183483

3-2 IPC 관련 매개 변수

IPC 관련 커널 매개 변수는 /proc/sys/kernel 디렉터리에 들어 있다. 관련 값은 웹(web), WAS(web application server), 데이터베이스(database) 서버 등을 운영하는 경우 각각의 서버가 요구하는 사항에 따라 설정해야 한다. IPC 관련 매개 변수는 공유 메모리 관련 변수, 세마포 관련 변수, 스와핑(swapping) 관련 변수로 나눌 수 있다.

1) 공유 메모리 관련 변수

공유 메모리 관리에서 이해해야 하는 용어가 페이지(page)와 세그먼트(segment)이다. 페이지는 주소 공간을 일정한 크기로 나눈 블록(block)으로 최근 리눅스 배포판에서는 4KB를 사용한다. 세그먼트는 고정된 단위가 아니고 논리적으로 관련된 정보의 단위로 프로그램에서 분할해서 사용한다. 만약 특정 프로그램에서 세그먼트를 10MB로 할당해서 사용한다면 1개 세그먼트의 단위는 10MB가 된다. 이 항목에 들어가는 매개 변수에는 shmmx, shmmni, shmall이 있다. shmmx는 프로세스가 사용할 수 있는 공유 메모리의 최대 크기, shmmni는 시스템 전체에서 사용할 수 있는 공유 메모리 세그먼트의 최대 개수, shmall은 모든 프로세스가 사용할 수 있는 공유 메모리 페이지의 최대 개수이다.

2) 세마포 관련 변수

시스템 내에 서버와 클라이언트가 존재할 경우 양단간의 통신 버퍼로 공유 메모리를 사용하는데, 이 때 메모리에 대한 동시성을 제어할 필요가 생긴다. 이 경우에 메모리에 대한 읽기 및 쓰기에 대한 접근권한의 제어를 위해 세마포 오퍼레이션(semaphore operation)이 발생한다. 세마포 오퍼레이션에 따라 프로세스는 대기상태가 되거나 진행상태가 되는데, 이러한 세마포 오퍼레이션은 동시다발적으로 발생되기 때문에 세마포의 개수 및 오퍼레이션에 대한 적절한 커널 매개 변수 값 설정이 중요하다. 세마포 관련 매개 변수에는 sem이 있는데, 이 변수 안에는 SEMMSL, SEMMNS, SEMOPM, SEMMNI와 같이 4개의 설정값이 들어 있다. SEMMSL은 한 개의 세마포 세트(배열)에 존재하는 세마포의 최대 개수, SEMMNS는 전체 시스템의 최대 세마포 수, SEMOPM은 세마포 콜(semop)에서 처리하는 최대 운영(operation) 수, SEMMNI는 세마포 세트(배열)의 수를 의미한다.

표 3. 공유 메모리 관련 커널 매개 변수
 Table 3. Shared memory related kernel parameters

parameter	default value
shmmx	18446744073692774399
shmmni	4096
shmall	18446744073692774399

표 4. 세마포 관련 커널 매개 변수

Table 4. Semaphore related kernel parameters

parameter	default value
sem	250 32000 32 128

표 5. 스와핑 관련 커널 매개 변수

Table 5. Swapping related kernel parameters

parameter	default value
swappiness	30

3) 스와핑(swapping) 관련 변수

운영체제에서는 성능적인 향상을 위해 사용되는 기술로 스와핑(swapping)과 캐시(file cache)를 손꼽을 수 있다. 리눅스에서는 이 두 기술과 연관된 커널 매개 변수를 /proc/sys/vm 디렉터리에 저장하고 있다. 먼저 스와핑은 주기억장치보다 더 큰 응용프로그램이나 데이터 파일을 처리하기 위해 디스크의 일부를 메모리처럼 사용하는 기술이다. 리눅스에서는 swappiness라는 커널 매개 변수를 제공하는데, 이 값의 수치를 조절하여 스와핑 발생 여부를 제어할 수 있다. swappiness는 스왑 성향을 수치로 정의하는 범위가 0부터 100까지이다. 이 값은 작을수록 적은 량의 스왑이 발생한다.

4) 캐시(cache) 관련 변수

캐시는 주기억장치와 보조기억장치간의 속도 차이로 인한 병목 현상을 해소하기 위해 운영체제에서 관리하는 일종의 시스템 버퍼이다. 리눅스 커널은 메모리 공간에 여유가 있을 때 cache를 할당하여 처리 속도를 높인다. cache로 할당되는 항목에는 page cache, inode cache, dentry cache가 있다. page cache는 시스템에서 처리되는 페이지가 저장되고, inode cache는 파일이나 디렉터리 정보를 담고 있는 inode table 관련 정보가 저장된다. 마지막으로 dentry는 directory-entry의 약자로 디렉터리 내의 요소들인 파일, 하위 디렉터리의 이름을 말한다. 각각의 파일 및 하위 디렉터리에 대한 정보는 inode 영역에 저장되기 때문에 해당 inode를 알아낼 수 있는 inode 번호와 함께 저장된다. 이러한 정보들은 파일 시스템 영역의 디스크 블록에 저장되어 있지만 성능 향상을 위해 메모리에 저장하는 데 이 영역을 dcache(dentry cache)라고 한다.

표 6. 캐시 관련 커널 매개 변수

Table 6. Cache related kernel parameters

parameter	default value
drop_caches	0

```
[root@localhost ~]# free -h
              total        used         free       shared  buff/cache   available
Mem:           1.8G         484M         529M          10M         826M         1.1G
Swap:          4.0G           0B           4.0G
[root@localhost ~]# cat /proc/sys/vm/drop_caches
0
[root@localhost ~]# echo 3 >/proc/sys/vm/drop_caches
[root@localhost ~]# free -h
              total        used         free       shared  buff/cache   available
Mem:           1.8G         459M         1.2G          10M         173M         1.2G
Swap:          4.0G           0B           4.0G
[root@localhost ~]# cat /proc/sys/vm/drop_caches
3
[root@localhost ~]#
```

그림 7. free 명령어 사용 예

Fig. 7. Example of free command

리눅스 커널 2.6.16 버전부터는 cache로 사용되는 메모리 영역을 즉시 회수할 수 있는 커널 매개 변수를 제공하는데, 이 변수가 /proc/sys/vm/drop_caches이다. 이 매개 변수를 이용한 캐시 메모리 영역 회수할 때 시스템에 문제가 없는 경우에만 가능하고, 디스크에 쓰기 작업이 실행되기 위해 대기하고 있는 경우에는 회수할 수 없다. 이 경우에는 'sync'와 같은 메모리 동기화 명령을 수행한 후에 다시 실행해야 한다. 아울러 명령을 실행 시에 캐시로 사용된 메모리가 회수됨으로 회수가 필요한 시점에서 반복 실행해야 한다. 표 6을 보면 기본값이 0으로 설정되어 있지만 값을 1-3 사이로 설정해서 캐시 메모리 영역을 회수할 수 있다. 1로 설정하면 page cache를 제거하고, 2로 설정하면 inode 및 dentry cache를 제거한다. 마지막으로 3으로 설정하면 page, inode, dentry cache 모두 제거할 수 있다[9].

캐시 메모리의 회수는 free 명령어로 확인할 수 있는데, 그림 7은 커널 매개 변수 값을 변경한 후에 비교한 결과이다. buff/cache 항목의 값이 크게 낮아지고, free 항목의 수치값이 증가된 것을 확인할 수 있다.

3-3 네트워크 관련 매개 변수

네트워크 관련 커널 매개 변수는 /proc/sys/net 디렉터리에 위치하고 있다. 이 디렉터리의 하위에 있는 주요 디렉터리에는 core, ipv4, ipv6, netfilter 등이 있다. core는 네트워크 기본 설정과 관련된 매개 변수가 들어 있고, ipv4는 ipv4 관련 커널 매개 변수가 들어 있다. ipv4의 하위 디렉터리인 conf에는 all, default, enp0s3(네트워크 장치명) 등과 같은 디렉터리가 존재하는데 모든 장치에 적용되게 하려면 all 디렉터리 내에 있는 매개 변수에 설정한다. ipv6는 ipv6 관련 커널 매개 변수가 들어 있는데, 기본적인 구조는 ipv4 디렉터리와 유사하다. netfilter에는 iptables 등과 같이 네트워크 필터링 관련 매개 변수가 들어 있다. 서버로 사용되는 경우가 많은 리눅스 운영체제의 특성상 네트워크 관련 커널 매개 변수가 상당히 많은 편이다. 본 논문에서는 네트워크 연결 상태와 관련된 값을 지정하는 수치형 매개 변수, 네트워크 관련 작업의 허가 여부를 결정하는 활성/비활성 매개 변수, 기타 매개 변수로 분류하여 분석하고, 주요 변수는 표 7부터 표 9와 같다.

```

root@localhost:~#
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost ~]# ls /proc/sys/net
core ipv4 ipv6 netfilter nf_conntrack_max unix
[root@localhost ~]#
    
```

그림 8. /proc/sys/net 디렉터리
 Fig. 8. /proc/sys/net directory

표 7. 네트워크 관련 수치형 커널 매개 변수
 Table 7. Network related numeric kernel parameters

parameter	default value
ipv4/tcp_keepalive_time	7200
ipv4/tcp_max_syn_backlog	128
ipv4/tcp_fin_timeout	60
ipv4/tcp_synack_retries	5
ipv4/tcp_max_tw_buckets	8192
ipv4/tcp_max_syn_backlog	128
ipv4/tcp_syn_retries	6
ipv4/tcp_retries2	15

표 8. 네트워크 관련 활성/비활성화 커널 매개 변수
 Table 8. Network related active/inactive kernel parameters

parameter	default value
ipv4/icmp_echo_ignore_all	0
ipv4/tcp_syncookies	1
ipv4/ip_forward	1
ipv4/icmp_echo_ignore_broadcasts	1
ipv4/icmp_echo_ignore_bogus_error_reponses	1
ipv4/tcp_timestamps	1
ipv4/conf/all/rp_filter	1
ipv4/conf/all/log_martians	0
ipv4/conf/*/accept_redirects	0
ipv4/conf/*/send_redirects	1
ipv4/conf/*/secure_redirects	1
ipv4/conf/*/accept_source_route	0

표 9. 네트워크 관련 기타 커널 매개 변수
 Table 9. Network related other kernel parameters

parameter	default value
ipv4/tcp_rmem	4096 87380 6291456
ipv4/tcp_wmem	4096 16384 6291456
ipv4/tcp_mem	42528 56704 85056
core/rmem_default	212992
core/wmem_default	212992
core/rmem_max	212992
core/wmem_max	212992
ip_local_port_range	32768 60999

IV. RHEL 6 및 7 커널 매개 변수 비교 분석

RHEL 7 버전에서 지원하는 세부 사양을 RHEL 6 버전과 비교해보면, 지원되는 파일 및 파일 시스템의 크기 등이 대폭 확장되고 커널의 기능도 강화되면서 리눅스 시스템의 성능도 크게 향상된 것을 알 수 있다. 이에 따라 리눅스 설치 시에 설정되는 커널 매개 변수의 기본값도 증가하고 있다. 커널 매개 변수의 기본값이 증가했다는 것은 리눅스 시스템에서 해당 부분의 성능이 좋아졌다고 볼 수 있다. 본 논문에서 인텔 i7 4세대 CPU가 장착된 시스템에 윈도우 10 운영체제를 설치하고, 버추얼박스(virtualbox) 기반의 가상화 환경에서 동일한 조건으로 2개의 가상 머신을 생성한 후에 RHEL 6 과 RHEL 7을 설치하여 커널 매개 변수 값을 비교해보았다.

4-1 파일 시스템 관련 매개 변수 비교

표 10을 보면 커널에서 최대 사용 가능한 파일의 값이 들어 있는 file-max의 기본수치가 RHEL 7이 RHEL 6보다 작을 것을 알 수 있다. 이것은 시스템의 성능이 저하되었다기보다는 해당 개수만큼 파일을 사용하는 경우가 없고, 시스템 내부에서 발생하는 있는 서비스 공격 거부에 대비해서 기본 수치를 낮춘 것으로 판단된다. 만약 동시에 사용하는 파일이 많다면 시스템의 상황에 맞게 해당 수치값을 높이면 된다.

표 10. 파일 시스템 관련 커널 매개 변수의 비교
 Table 10. Comparison of File system related kernel parameters

parameter	default value	
	RHEL 6	RHEL 7
file-max	203819	183483
file-nr	5504 0 203819	6592 0 183483

4-2 IPC 관련 매개 변수 비교

1) 공유 메모리 관련 변수 비교

리눅스 운영체제를 서버로 사용하고 동시에 운영되는 프로세스들이 많다면 시스템 성능과 가장 밀접한 영역이 공유 메모리 매개 변수이다. 표 11을 보면 수치가 엄청나게 늘어난 것을 알 수 있다. 특히 프로세스가 사용할 수 있는 공유 메모리의 최대 크기를 나타내는 `shmmax`의 수치가 엄청나게 늘어난 것을 알 수 있다. 해당 값의 단위는 byte인데, RHEL 7의 기본값은 약 16EB이다.

2) 세마포 관련 변수

표 12를 보면 세마포 관련 커널 매개 변수의 기본값 수치 변화는 없다. 해당 값만으로도 충분한 것으로 판단되는데, 필요하면 수동으로 수치값을 조절하면 된다.

3) 스와핑(swapping) 관련 변수

`swappiness`는 값은 작을수록 적은 양의 스왑이 발생하는데, 표 13을 보면 RHEL 7은 30으로 RHEL 6의 60에 비해 현저히 낮아진 것을 알 수 있다. 스왑은 물리적으로 메모리가 부족한 경우에는 유용하지만 스와핑을 많이 한다는 것은 시스템의 성능을 저하시킬 수 있다. 최근 리눅스 시스템에 장착되는 물리적 메모리의 양이 증가하는 추세를 반영한 것으로 판단된다.

표 11. 공유 메모리 관련 커널 매개 변수의 비교

Table 11. Comparison of Shared memory related kernel parameters

parameter	default value	
	RHEL 6	RHEL 7
shmmax	4294967295	18446744073692774399
shmmni	4096	4096
shmall	268435456	18446744073692774399

표 12. 세마포 관련 커널 매개 변수의 비교

Table 12. Comparison of Semaphore related kernel parameters

parameter	default value	
	RHEL 6	RHEL 7
sem	250 32000 32 128	250 32000 32 128

표 13. 스와핑 관련 커널 매개 변수의 비교

Table 13. Comparison of Swapping related kernel parameters

parameter	default value	
	RHEL 6	RHEL 7
swappiness	60	30

4-3 네트워크 관련 매개 변수 비교

표 7에서 제시한 네트워크 연결 상태와 관련된 변수 중에 기본값이 변경된 매개 변수만 정리해서 비교해보면 표 14과 같다. `tcp_max_syn_backlog`는 TCP 프로토콜에서 연결 요청이 아직 완전히 처리되지 아니한 대기 상태에 있는 큐의 길이를 나타낸다. TCP 프로토콜에서 하나의 소켓이 동시에 SYN 요청을 처리하기에는 한계가 있는데, 이 한계가 백로그(backlog)이고, 설정된 수치값에 해당하는 백로그의 큐가 꽉 차게 되면 이후 들어오는 SYN요청은 무시된다. `tcp_max_tw_buckets`는 TIME_WAIT을 담을 수 있는 소켓 수를 나타내고, `tcp_syn_retries`는 활성화된 TCP 접속이 재전송을 시도하기 위한 최초 SYN 시간의 값이다. RHEL 7에서는 RHEL 6과 비교했을 때 기본값을 크게 낮춘 것을 확인할 수 있다. 최근 서버로 운영되는 시스템에 대한 서비스 거부 공격이 많아지는 추세여서 시스템의 보안 강화를 변경한 것으로 판단된다. 시스템 관리자는 시스템 상황에 맞게 수치를 조절하면 된다.

표 9에 제시된 네트워크 관련 기타 매개 변수의 대부분은 네트워크 송수신 버퍼 크기와 관련된 커널 매개 변수이다. `ipv4/tcp_rmem`은 TCP 레이어 수신 버퍼값이고, `ipv4/tcp_wmem`은 TCP 레이어 송신 버퍼값으로 최소, 기본값, 최대값으로 기록되어 있다. `ipv4/tcp_mem`은 TCP stack이 메모리를 어떻게 사용할지에 대한 메모리 페이지 단위 설정이다. `core/rmem_default` 및 `core/wmem_default`는 네트워크의 읽기 및 쓰기 버퍼 크기이고, `core/rmem_max` 및 `core/wmem_max`는 최댓값을 의미한다. 표 15를 보면 RHEL 7에서는 RHEL 6과 비교했을 때 최대 지원되는 수치값은 2배로 늘렸지만, 기본값은 낮춘 것을 확인할 수 있다. 이 항목도 마찬가지로 성능 향상에 따라 지원되는 최댓값의 수치는 높고, 서비스 거부 공격에 대비해서 기본 수치는 낮춘 것으로 판단된다.

표 14. 네트워크 관련 커널 매개 변수 비교

Table 14. Comparison of Network related kernel parameters

parameter	default value	
	RHEL 6	RHEL 7
ipv4/tcp_max_syn_backlog	512	128
ipv4/tcp_max_tw_buckets	65536	8192
ipv4/tcp_syn_retries	5	6

표 15. 네트워크 관련 기타 커널 매개 변수

Table 15. Network related other kernel parameters

parameter	default value	
	RHEL 6	RHEL 7
ipv4/tcp_rmem	4096 87380 3424256	4096 87380 6291456
ipv4/tcp_wmem	4096 16384 3424256	4096 16384 6291456
ipv4/tcp_mem	80256 107008 160512	42528 56704 85056
core/rmem_default	112640	212992
core/wmem_default	112640	212992
core/rmem_max	112640	212992
core/wmem_max	112640	212992

V. 결론

리눅스는 소스가 공개된 운영체제의 특성처럼 시스템의 성능을 향상시키거나 보안을 강화할 수 있는 다양한 커널 매개 변수를 제공하여 시스템 관리자가 사용하는 환경에 맞게 최적화시킬 수 있다. 특히 파일 시스템 관련 매개 변수, IPC와 관련된 공유 메모리 및 세마포 관련 매개 변수, 스왑 및 페이지 같은 메모리 관련 매개 변수, TCP 및 ICMP 등의 네트워크 관련 매개 변수 등은 시스템 성능 향상과 보안 강화에 큰 역할을 수행할 수 있다. 리눅스 커널 버전이 올라가고 리눅스 배포판의 버전도 올라가면서 운영체제로서의 성능이 향상되고 있고, 이에 따라 지원되는 커널 매개 변수의 최댓값도 크게 상승하고 있다. 현재 리눅스 운영체제는 데스크톱이나 서버뿐만 아니라 사물인터넷 기기 등 다양한 분야에서 사용되고 있는데, 각각의 분야에서 요구하는 사항을 분석하여 상황에 맞게 리눅스 커널 매개 변수값을 조정한다면 리눅스 시스템의 성능을 향상시킬 수 있을 뿐만 아니라 보안성 강화에도 큰 기여를 할 것으로 판단된다.

참고문헌

[1] S. J. Jung, and K. Sung, "Trend analysis and Classification of Linux distributions," *The Journal of Digital Contents Society*, Vol. 18, No. 2, pp. 357-363, April 2017.

[2] S. J. Jung, *Linux Master Second Class Complete Conquest*, Seoul, Booksholic publishing, 2017.

[3] S. J. Jung, B. Y. Bae, B. W. Shin, J. S Park, J. H. Shin, and K. Y. Lee, *Understanding the Linux Core with CentOS7*. Seoul, Booksholic publishing, 2018.

[4] K. Sung, "Trend analysis of X Window used in Linux," *The Journal of Digital Contents Society*, Vol. 18, No. 7, pp.

1393-1401, November 2017.

[5] K. Sung, "Trend analysis of Linux System changes by adoption XFS File System," *The Journal of Digital Contents Society*, Vol. 19, No. 3, pp. 497-503, March 2018.

[6] S. J. Jung, and Y. M. Bae, *To Conquer the Linux Master First Class*. Seoul, Booksholic publishing, 2015.

[7] Wikipedia, Inter-process Communication [Internet]. Available: https://en.wikipedia.org/wiki/Inter-process_communication/.

[8] Altibase, Technical Documents [Internet]. Available: <http://docs.altibase.com/pages/viewpage.action?pageId=10846396>.

[9] Linux Memory Management, Drop_Caches [Internet]. Available: https://linux-mm.org/Drop_Caches.

배유미(Yu-Mi Bae)



2007년 : 한남대학교 대학원 (공학석사)
2013년 : 한남대학교 대학원 (공학박사)

2015년~2018년: (주)엔버 기업부설연구소 선임연구원
2018년~현 재: 한남대학교 탈메이지교양대학 초빙교수
※관심분야: 리눅스, 정보보안, 멀티미디어, 클라우드 컴퓨팅

박정수(Jeong-Su Park)



2014년 : 숭실대학교 대학원 (공학석사)

2013년~현 재: 코오롱베니트(주) 과장
※관심분야: 리눅스, 운영 체제, 클라우드 컴퓨팅, SW테스트, 자동화

신병웅(Byung-Woong Shin)



2004년 : 목원대학교 컴퓨터공학과 (공학사)

2017년~현 재: DellEMC senior systems engineer (부장)
※관심분야: 리눅스, 운영 체제, 클러스터, 가상화, 클라우드

정성재(Sung-Jae Jung)



2003년 : 한남대학교 대학원 (공학석사)
2011년 : 한남대학교 대학원 (공학박사)

2005년~2010년: 한남대학교 국제IT교육센터 전임강사
2015년~현 재: (주)엔버 기업부설연구소장
※관심분야: 리눅스, 정보보안, 시스템보안, 클라우드 컴퓨팅 등