

Admin : New post

꿈을 꾸지 않는자는 살아있는 송장. 난 차라리 꿈꾸다 죽은 시체가 되길. 똑같은 오늘날을 찍어내는 공장에 사느니 난 차라리 새로운 내일에 죽으리. 꿈을 꾸는건, 약속을 하는 것. 내일의 자신에게 보내는 편지를 쓰는 것. <다이나믹 듀오 - Dream> 중에서..



CATEGORY

- T 분류 전체보기 (728)
  - Movie Story (106)
  - Animation Story (5)
  - Monolog (62)
  - Drama Story (18)
  - Music makes me high (27)
  - Fun / Information (62)
  - Secret Scrap (2)
  - Daily (11)
  - Food and Place (3)
  - Embedded Device (269)
  - Desktop (101)
  - Programming (58)
  - Dev. Note (Private) (0)
  - Study. Note (4)

미디어로그

NOTICE  
RECENT ENTRIES  
RECENT COMMENTS  
RECENT TRACKBACKS  
CALENDAR  
VISITOR STATISTICS

-Total : 434,471  
-Today : 110  
-Yesterday : 202

TAG

영화 쏘우 영화 배트맨 virtual box 영화 트랜스포터 영화 스파이더맨 영화 엑스맨 USB 드라마 로스트 드라마 히어로즈 영화 헬보이

GET RSS PAGE RSS 2.0  
꽃피는 봄나들이 in bloom (flower)  
SKIN BY WHITED/LonnieNa

TISTORY

티스토리 가입하기!

[Linux] 메모리 사용량 조사 방법 + [Desktop/Linux] | 2014.05.28 10:57

Creative Commons License



이 저작물은 크리에이티브 커먼즈 코리아 저작자표시-비영리-동일조건변경허락 2.0 대한민국 라이선스에 따라 이용하실 수 있습니다.

출처: <http://blog.naver.com/PostView.nhn?blogId=jwmoon74&logNo=100174011942>

## 특정 프로세스의 메모리 사용량(memory usage) 확인하는 방법

- /proc/프로세스ID/stat 또는 /proc/프로세스ID/statm 파일에 접근하여 확인이 가능함
- man proc으로 자세한 필드 정보를 확인할 수 있음
- stat의 vsize, rss 값을 참고하면 됨
- /proc/self를 통해 자신의 프로세스 ID를 구하지 않고도 접근이 가능함

/proc/프로세스아이디/status 에 기록되어있음.

VmSize : 전체 할당된 가상 메모리의 크기  
VmRSS : 실제 물리적 메모리가 할당된 크기  
VmLck : 스왑아웃 필수 없는 메모리의 크기  
다음은 어떠한 용도로 사용되는냐에 따라서 나눈 것입니다.  
VmData: Heap 영역  
VmStk : Stack 영역  
VmExe : 실행코드 영역 (전역변수 및 실행코드)  
VmLib : 동적으로 연결된 라이브러리 영역

C 코드로 정보 가져오기 예제)

```
1 char line[1024];
2
3 sprintf(line, "/proc/%d/status", getpid());
4 FILE * fp = fopen(line, "r");
5 if (fp == NULL)
6     return;
7
8 int nVmSize = 0;
9 int nVmRss = 0;
10
11 while (fgets(line, 1024, fp) != NULL) {
12     if (strstr(line, "VmSize"))
13     {
14         char tmp[32];
15         char size[32];
16         sscanf(line, "%s%s", tmp, size);
17         nVmSize = atoi(size);
18     }
19     else if (strstr(line, "VmRSS")) {
20         char tmp[32];
21         char size[32];
22         sscanf(line, "%s%s", tmp, size);
23         nVmRss = atoi(size);
24         break;
25     }
26 }
27
28 fclose(fp);
```

## 명령어 : free

free명령어는 시스템의 실제메모리와 스왑메모리에 대한 사용현황을 확인할 수 있는 명령어이다

free명령어로 확인할 수 있는 정보

- \* 실제메모리의 전체용량
- \* 실제메모리 중 유휴메모리의 량
- \* 실제메모리 중 사용중인 메모리의 량
- \* 스왑메모리의 량
- \* 커널에서 사용되는 공유메모리와 버퍼의 량
- \* 캐시된 메모리의 량

예제

```
[root@Client ~]# free
              total        used         free       shared    buffers     cached
Mem:      255596      238328         17268           0        19352      128608
-/+ buffers/cache:      98376        157220
Swap:      522872           0         522872
```

free명령어의 옵션

- b : byte단위로 표시
- k : kilo byte단위로 표시(default)
- m : mega byte단위로 표시
- t : 맨 마지막에 total용량을 표시
- o : buffer/cache 라인을 표시하지 않는다
- s : 지정한 초단위로 refresh하여 메모리상태를 표시

-V : free명령어의 버전을 표시

### 각 항목들의 의미

Mem행 : 시스템의 물리적인 메모리에 대한 사용량을 각 필드로 단위로 표시

total : 전체 메모리의 용량을 Kbyte단위(default)로 표시  
used : 현재 시스템에서 사용중인 메모리의 량을 Kbyte단위로 표시  
free : 현재 시스템에서 사용중이지 않은 메모리의 량을 Kbyte단위로 표시  
shared : 현재 시스템에서 공유한 메모리의 용량을 표시  
buffers : 현재 시스템에서 buffering된 메모리의 량을 표시  
cached : 현재 시스템에서 caching된 메모리의 량을 표시

-/+ buffers/cache : 현재 캐시메모리에서 버퍼링된 사용량을 표시

Swap행 : 서버설치시에 결정한 스왑메모리의 량, 스왑메모리는 디스크의 일부분을 메모리로 잡아서  
설정되기 때문에 스왑메모리가 많이 사용되고 있다는 것은 시스템의 전체적인 속도가 떨어진다는 것을 의미하며 지속적으로 스왑메모리가 사용된다는 것은 결국 실제메모리를 증설해야 한다는 것이다.

total : 시스템의 전체 스왑메모리의 량을 표시  
used : 전체 스왑메모리중에서 현재 사용중인 스왑메모리의 량을 표시  
free : 전체 스왑메모리중에서 사용되지 않고 남아 있는 메모리의 량

## 명령어 : top

리눅스는 실행중인 어플리케이션에서 요구하는 메모리를 제외한 대부분의 메모리를 디스크 캐쉬로 활용한다. 이러한 사실을 모르고 있다면, 메모리 현황 조회시 왜 free 메모리가 부족한지를 이해할 수 없을 것이다.

사실은 내가 그랬다. ^^;

Linux 에서는 메모리 사용 현황을 **top 명령**으로 조회할 수 있으며, 아래는 top 결과 샘플이고 샘플 내의 여러 항목들 중에서 free 와 cached 의 값이 이번 주제에서 중요 항목이다.

```
top - 09:40:42 up 74 days, 16:47, 3 users, load average: 0.00, 0.02, 0.08
Tasks: 212 total, 1 running, 210 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.0% us, 0.0% sy, 0.0% ni, 99.9% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 16630888k total, 16559108k used, 71780k free, 100516k buffers
Swap: 16771776k total, 31920k used, 16739856k free, 16034200k cached
```

free 메모리가 작지만, cached 메모리는 어플리케이션이 메모리를 필요로 할 때 바로 반환될 수 있으므로 **cached 메모리를 실질적으로는 free 메모리로 보아도 무방하다.**

따라서 리눅스에서 **가용 메모리 계산은 free + buffers + cached 로 할 수 있다.**

위의 top 명령 결과로 메모리 용량을 분석해 본다면,  
- 전체 Physical 메모리 : 16630888k total  
- 실제 사용중인 메모리 (used - cached - buffers) : 16559108k - 16034200k - 100516k = 424392k  
- 실제 가용한 메모리 (free + buffers + cached) : 71780k + 100516k + 16034200k = 16206496k  
- 전체 메모리 (실제 사용중인 메모리 + 실제 가용한 메모리) : 424392k + 16206496k = 16630888k

아래는 **free -m** 명령으로 조회한 결과이다. (m 옵션은 MB 단위 출력 지정)

	total	used	free	shared	buffers	cached
Mem:	16241	16188	52	0	98	15669
-/+ buffers/cache:		419	15821			
Swap:	16378	31	16347			

Mem: 라인에서의 free + buffers + cached 의 값은 -/+ buffers/cache: 라인의 free 값과 비슷하다.

## [Linux] Top 명령어 사용법

Linux 시스템을 사용하다가 시스템 사용량을 보기위해 우리는 top를 이용한다  
top 명령어 사용시 유용한 옵션정리

**[사용법]**  
**\$ top [옵션]**

나의 경우는 옵션을 별로 사용하지 않는다  
보통 top 화면으로 들어가서 사용량을 확인함

### [화면 내용들]

```
top - 15:55:22 up 98 days, 14:29, 1 user, load average: 0.05, 0.04, 0.00
Tasks: 141 total, 1 running, 140 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.2%us, 0.1%sy, 0.0%ni, 99.3%id, 0.4%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 4031448k total, 3306440k used, 725008k free, 263928k buffers
Swap: 8193140k total, 2900k used, 8190240k free, 1588424k cached
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
```

### 각 라인별 내용

**top - 15:55:22 up 98 days, 14:29, 1 user, load average: 0.05, 0.04, 0.00**

현재 시간, uptime (서버 uptime), 현재 로그인 유저수 , 로드 에버리지 1분 5분 15분

### 로드 에버리지(load average)란?

- 작업의 대기시간, 값이 1이라면 1분동안 평균 1개의 프로세스가 대기상태임을 나타낸다.  
보통 5이면 서버가 부하를 받는다고 생각함, 10~15면 과부하

Tasks: 141 total, 1 running, 140 sleeping, 0 stopped, 0 zombie

전체 프로세스 수, 현재 실행중인 프로세스, 유휴상태 프로세스, 정지상태 프로세스, 좀비 프로세스

Cpu(s): 0.2%us, 0.1%sy, 0.0%ni, 99.3%id, 0.4%wa, 0.0%hi, 0.0%si, 0.0%st

사용자가 사용중인 CPU 사용률(us), 시스템이 사용하는 CPU 사용률(sy), NICE 정책에 의해 사용되는 CPU 사용률(ni), 사용되지 않은 CPU의 미사용률(id), 입출력 대기상태의 사용률(wa)

Mem: 4031448k total, 3306440k used, 725008k free, 263928k buffers

전체 물리적인 메모리, 사용중인 메모리(used), 사용되지 않는 여유 메모리(free), 버퍼된 메모리(buffers)

Swap: 8193140k total, 2900k used, 8190240k free, 1588424k cached

전체 스왑 메모리, 사용중인 스왑 메모리, 남아있는 스왑메모리, 캐싱메모리

### 세부정보 필드명

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

\* PID : 프로세스 ID (PID)

\* USER : 프로세스를 실행시킨 사용자 ID

\* PRI : 프로세스의 우선순위 (priority)

\* NI : NICE 값. 일의 nice value값이다. 마이너스를 가지는 nice value는 우선순위가 높음.

\* VIRT : 가상 메모리의 사용량(SWAP+RES)

\* RES : 현재 페이지가 상주하고 있는 크기(Resident Size)

\* SHR : 분할된 페이지, 프로세스에 의해 사용된 메모리를 나눈 메모리의 총합.

\* S : 프로세스의 상태 [ S(sleeping), R(running), W(swapped out process), Z(zombies) ]

\* %CPU : 프로세스가 사용하는 CPU의 사용율

\* %MEM : 프로세스가 사용하는 메모리의 사용율

\* COMMAND : 실행된 명령어

### [명령 실행 후 사용하는 옵션들]

1 : cpu 갯수별 사용량 보기

shift + m : 메모리 사용량이 큰 순서로 정렬

shift + p : CPU 사용량이 큰 순서로 정렬

shift + t : 실행시간이 큰 순서로 정렬

k : 프로세스 kill - k 입력 후 종료할 PID 입력 signal을 입력하라고 하면 kill signal인 9를 입력

c : 명령인자 표시/비표시

space bar : refresh

u : 입력한 유저의 프로세스만 표시 - which user 가 나오면 유저를 입력, 아무것도 안넣으면 전체 표시

shift + b : 상단 정보를 블록표시상태로 보여줌

공감

신고

이웃

TRACKBACK 0 COMMENT 0

ID

PW

HOME

http://

비밀글로 저장

1234567890

WRITE

<<이전 | 1 ... | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 ... | 728 | 다음>>

rednine's Blog is powered by Daum & tistory