

Лабораторна робота № 3

НАЗВА: Моніторинг використання оперативної пам'яті в ОС Linux.

МЕТА: Навчитись роботі з командами та системними утилітами моніторингу використання оперативної пам'яті в ОС Linux.

1. Загальні відомості

1.1. Детальну інформацію про біжучий розподіл та використання оперативної пам'яті можна отримати за допомогою віртуальної файлової системи procfs (/proc file system), виконавши в командному рядку:

```
$ cat /proc/meminfo
```

Параметри використання пам'яті у віртуальному файлі **/proc/meminfo** визначаються в режимі реального часу в момент звертання до вмістимого файлу. Основні параметри:

MemTotal - загальний об'єм доступної оперативної пам'яті (тобто фізичний об'єм RAM мінус кілька зарезервованих бітів і мінус бінарний код ядра);

MemFree - об'єм вільної пам'яті;

MemAvailable - оцінка об'єму пам'яті доступної для запуску нової програми (за виключенням свопінгу);

Buffers - пам'ять, що використовується для буферизації файлів (тимчасового зберігання блоків диску);

Cached - розмір кешу для файлів, які читаються з диску (pagescache), за виключенням SwapCached;

SwapCached - об'єм даних які одночасно знаходяться в оперативній пам'яті і у свопі (для економії часу на операціях вводу/виводу);

SwapTotal - загальний доступний об'єм свопу (swap);

SwapFree - об'єм вільної пам'яті у свопі.

Приклади:

1) визначити біжучий сумарний об'єм спільної пам'яті (shared memory) у системі:

```
$ cat /proc/meminfo | grep Shmem:
```

```
Shmem:          327188 kB
```

2) вивести основні параметри використання оперативної пам'яті і свопу:

```
$ egrep --color 'Mem|Buffers|Cached|Swap' /proc/meminfo
```

```
MemTotal:       3955628 kB
MemFree:        733420 kB
MemAvailable:   1875064 kB
Buffers:        119776 kB
Cached:         1509596 kB
SwapCached:      0 kB
SwapTotal:      4099068 kB
SwapFree:       4099068 kB
```

1.2. Команда **free** призначена для отримання інформації про оперативну пам'ять (загальний об'єм, об'єм використаної пам'яті, об'єм вільної пам'яті та ін.) у зручному вигляді з можливістю

уточнення інформації за допомогою опцій команди. Інформація, яку виводить команда **free**, береться з віртуального файлу /proc/meminfo.

Приклади:

1) вивести інформацію про використання пам'яті у зручному вигляді:

```
$ free -h
```

2) вивести підсумкову інформацію про використання пам'яті:

```
$ free -ht
```

3) виводити інформацію про використання пам'яті кожні 5 секунд:

```
$ free -s 5
```

1.3. Системна утиліта **vmstat** (virtual memory statistics) входить до набору утиліт sysstat (system statistics) і призначена для отримання інформації про віртуальну пам'ять системи. За допомогою утиліти **vmstat** можна організувати збір статистичної інформації про різні аспекти використання оперативної пам'яті.

Приклади:

1) вивести інформацію про використання оперативної пам'яті:

```
$ vmstat
```

```
procs -----memory----- ---swap-- ----io---- -system-- -----cpu-----
 r b   swpd   free   buff   cache   si   so    bi    bo    in   cs us sy id wa st
 0 0       0 934748 121036 1454304    0    0    27    11   111  607  3  1 95  1  0
```

2) виводити інформацію про біжуче використання оперативної пам'яті кожні 5 секунд 10 разів:

```
$ vmstat 5 10
```

3) виводити інформацію про біжуче використання оперативної пам'яті з часовими відмітками:

```
$ vmstat -t 1 5
```

```
procs -----memory----- ---swap-- ----io---- -system-- -----cpu----- ----timestamp-----
 r b   swpd   free   buff   cache   si   so    bi    bo    in   cs us sy id wa st      EET
 2 0       0 949780 121180 1453080    0    0    26    11   110  601  3  1 95  1  0 2019-11-28 14:52:17
 0 0       0 945688 121180 1457076    0    0    0     0    101  326  0  0 99  0  0 2019-11-28 14:52:18
 0 0       0 945776 121180 1457112    0    0    0     0    107  371  0  0 99  0  0 2019-11-28 14:52:19
 2 0       0 945156 121180 1457636    0    0    0     0    82  299  0  0 99  0  0 2019-11-28 14:52:20
 0 0       0 945156 121180 1457636    0    0    0     0   392 1662  1  0 99  0  0 2019-11-28 14:52:21
```

4) вивести статистичну інформацію про використання оперативної пам'яті та системні події:

```
$ vmstat -s
```

1.4. Команда **mpstat** призначена для отримання інформації про використання пам'яті окремим обчислювальним процесом або набором процесів. Команда **mpstat** дозволяє визначити які об'єкти даних знаходяться в пам'яті процесу, де вони розміщені і який об'єм займають (memory map).

Приклади:

1) вивести інформацію про пам'ять процесу з ідентифікатором 3901:

```
$ pmap 3901
```

```
3901:  bash
0000000000400000    1024K r-x-- bash
0000000000700000     12K r---- bash
0000000000703000     36K rw--- bash
000000000070c000     40K rw--- [ anon ]
0000000000859000    1752K rw--- [ anon ]
00007f8233021000     40K r-x-- libnss_files-2.24.so
00007f823302b000    2048K ----- libnss_files-2.24.so
00007f823322b000     4K r---- libnss_files-2.24.so
00007f823322c000     4K rw--- libnss_files-2.24.so
00007f823322d000     24K rw--- [ anon ]
00007f8233233000     44K r-x-- libnss_nis-2.24.so
00007f823323e000    2044K ----- libnss_nis-2.24.so
00007f823343d000     4K r---- libnss_nis-2.24.so
00007f823343e000     4K rw--- libnss_nis-2.24.so
00007f823343f000     80K r-x-- libnsl-2.24.so
00007f8233453000    2048K ----- libnsl-2.24.so
00007f8233653000     4K r---- libnsl-2.24.so
00007f8233654000     4K rw--- libnsl-2.24.so
. . .
00007ffff42db9000     8K r-x-- [ anon ]
fffffffffff600000     4K r-x-- [ anon ]
total                21284K
```

2) вивести інформацію про пам'ять процесу з ідентифікатором 3901 в розширеному вигляді:

```
$ pmap -x 3901
```

```
3901:  bash
Address      Kbytes      RSS      Dirty Mode  Mapping
0000000000400000    1024      948          0 r-x--  bash
0000000000400000         0         0          0 r-x--  bash
0000000000700000         12        12          4 r----  bash
0000000000700000         0         0          0 r----  bash
0000000000703000         36        36         36 rw---  bash
0000000000703000         0         0          0 rw---  bash
000000000070c000         40        28         28 rw---  [ anon ]
000000000070c000         0         0          0 rw---  [ anon ]
0000000000859000    1752    1748    1748 rw---  [ anon ]
0000000000859000         0         0          0 rw---  [ anon ]
00007f8233021000         40        40          0 r-x--  libnss_files-2.24.so
00007f8233021000         0         0          0 r-x--  libnss_files-2.24.so
00007f823302b000    2048         0          0 ----- libnss_files-2.24.so
. . .
fffffffffff600000         4         0          0 r-x--  [ anon ]
fffffffffff600000         0         0          0 r-x--  [ anon ]
-----
total kB          21284    5188    1972
```

3) вивести інформацію про пам'ять декількох процесів:

```
$ pmap 3901 5051 5645
```

1.5. Системна утиліта **smem** призначена для формування звітів про використання оперативної пам'яті обчислювальними процесами з врахуванням сторінок спільної пам'яті (shared memory). Утиліта **smem** виводить інформацію про три показники використання пам'яті процесом: USS (Unique Set Size) – об'єм пам'яті процесу без врахування спільної пам'яті (значення USS вказує скільки пам'яті буде вивільнено в разі завершення виконання процесу); PSS (Proportional Set Size) – об'єм пам'яті процесу з додатковим врахуванням спільної пам'яті (до USS додається пропорційна частка спільної пам'яті даного процесу з урахуванням кількості інших процесів, які розділяють цю область спільної пам'яті); RSS (Resident Set Size) – об'єм частки оперативної пам'яті, яку займає процес (оцінка об'єму пам'яті необхідного для виконання процесу).

Приклади:

1) отримати інформацію про використання пам'яті процесами:

\$ smem

PID	User	Command	Swap	USS	PSS	RSS
1200	smm	/usr/bin/dbus-daemon --conf	0	456	600	3828
1125	smm	/usr/lib/gdm3/gdm-x-session	0	672	743	5528
1195	smm	/usr/lib/at-spi2-core/at-sp	0	684	801	6272
1328	smm	/usr/lib/gvfs/gvfs-mtp-volu	0	780	856	5576
1336	smm	/usr/lib/gvfs/gvfs-goa-volu	0	756	870	5640
1458	smm	/usr/lib/dconf/dconf-servic	0	916	972	4936
1220	smm	/usr/lib/gvfs/gvfsd	0	880	1042	6468
1202	smm	/usr/lib/at-spi2-core/at-sp	0	908	1053	6964
1611	smm	/usr/lib/gvfs/gvfsd-burn --	0	896	1057	6428
1225	smm	/usr/lib/gvfs/gvfsd-fuse /r	0	1072	1145	5964

...

2) отримати інформацію про використання пам'яті процесами firefox:

\$ smem --processfilter="firefox"

PID	User	Command	Swap	USS	PSS	RSS
6052	smm	/usr/bin/python /usr/bin/sm	0	13712	13749	15956
3136	smm	/home/smm/firefox/firefox-b	0	24396	32145	89148
4464	smm	/home/smm/firefox/firefox-b	0	55692	70865	155804
3857	smm	/home/smm/firefox/firefox-b	0	67496	83174	170096
3784	smm	/home/smm/firefox/firefox-b	0	74932	91103	178160
3110	smm	/home/smm/firefox/firefox-b	0	96488	114267	205436
3821	smm	/home/smm/firefox/firefox-b	0	121372	151881	254768
2997	smm	/home/smm/firefox/firefox	0	230792	270512	381732

3) вивести узагальнену інформацію про використання пам'яті в системі:

\$ smem -w

Area	Used	Cache	Noncache
firmware/hardware	0	0	0
kernel image	0	0	0
kernel dynamic memory	1204476	1061764	142712
userspace memory	1795672	565528	1230144
free memory	955480	955480	0

4) вивести інформацію про використання пам'яті в системі зведену по користувачам:

\$ sudo smem -u

User	Count	Swap	USS	PSS	RSS
daemon	1	0	232	266	2296
rtkit	1	0	416	463	3148
systemd-timesync	1	0	600	925	4772
avahi	2	0	400	954	5376
Debian-exim	1	0	1316	1369	3108
messagebus	1	0	1464	1615	4844
colord	1	0	5276	5742	13500
root	26	0	60428	67266	179372
smm	61	0	1500180	1727221	2937784

2. Послідовність виконання роботи

2.1. Ознайомитись з відомостями про моніторинг використання оперативної пам'яті в ОС Linux.

2.2. Дослідити вміст віртуального файлу /proc/meminfo. Визначити за його допомогою біжучий сумарний об'єм спільної пам'яті (shared memory) у системі та вивести основні параметри використання оперативної пам'яті і своєї.

2.3. Дослідити роботу команди free. Вивести командою free інформацію про використання пам'яті в системі.

2.4. Дослідити роботу системної утиліти `vmstat`. Отримати за її допомогою 1) інформацію про використання оперативної пам'яті, 2) статистичну інформацію про використання оперативної пам'яті та системні події.

2.5. Дослідити роботу команди `mpar`. Отримати за її допомогою інформацію про пам'ять процесу `bash` у звичайному та розширеному вигляді.

2.6. Дослідити роботу системної утиліти `smem`. Отримати за її допомогою 1) інформацію про використання пам'яті процесами, 2) узагальнену інформацію про використання пам'яті в системі, 3) інформацію про використання пам'яті в системі зведену по користувачам.

2.7. Скласти та захистити звіт з лабораторної роботи.

3. Зміст звіту

3.1. Результати виконання завдань по дослідженню вмісту віртуального файлу `/proc/meminfo`.

3.2. Результати дослідження роботи команди `free`.

3.3. Результати виконання завдань по дослідженню роботи утиліти `vmstat`.

3.4. Результати виконання завдань по дослідженню роботи команди `mpar`.

3.5. Результати виконання завдань по дослідженню роботи утиліти `smem`.

4. Контрольні питання

4.1. Віртуальний файл `/proc/meminfo`.

4.2. Призначення та використання команди `free`.

4.3. Призначення та використання системної утиліти `vmstat`.

4.4. Призначення та використання команди `mpar`.

4.5. Призначення та використання системної утиліти `smem`.

5. Джерела

1. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd edition, O'Reilly Media, 2005. - 944 p.

2. Robert Love, Linux Kernel Development, 3rd edition, Addison-Wesley Professional, 2010. - 440 p.

3. The Linux Kernel documentation, www.kernel.org/doc/html/latest/

4. The `/proc` file system (Linux Documentation), <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation/filesystems/proc.txt>