**Лабораторная работа №5. Управление памятью в ОС Linux**

**Данные о текущей конфигурации операционной системы:**

* **MemTotal: 1.860.932 kB**
* **SwapTotal: 839.676 kB**
* **PageSize: 4096 bytes**
* **MemFree: 1.463.043 kB**
* **SwapFree: 578.676 kB**

**Эксперимент №1**

**I этап (Run1)**

Задача – оценить изменения параметров, выводимых утилитой top в процессе работы созданного скрипта.

1. Аварийная остановка процесса **run1.bash:**
   1. **Системный журнал:**

**[ 9004.636739] Out of memory: Killed process 20168 (bash) total-vm:2401264kB, anon-rss:1517828kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:4312kB oom\_score\_adj:0  
[ 9004.909662] oom\_reaper: reaped process 20168 (bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB**

* 1. **report.log:**

**<…>**

**29000000**

**30000000**

1. Результаты замеров c использованием **follower.bash** из **run1.bash**:
2. Выводы:

Вначале мы видим постепенный рост использования оперативной памяти, при этом раздел подкачки остаётся нетронутым. Как только процесс использует всю доступную ему память, начинается процесс страничного обмена, это заметно по колебаниями использования процессорного времени – процесс обмена забирает часть ресурсов. Начинается расход памяти в подкачке, процесс обменивает страницы на свободные в подкачке, заполняет их и снова производит обмен. Как только вся оперативная память вместе с подкачкой исчерпана – процесс завершается аварийно.

**II этап (Run2)**

1. Аварийная остановка процесса **run2.bash:**
   1. **Системный журнал:**

**[ 13281.174607] Out of memory: Killed process 30170 (bash) total-vm:1358860kB, anon-rss:836224kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:2272kB oom\_score\_adj:0  
[ 13281.310359] oom\_reaper: reaped process 30170 (bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB**

**[ 13336.460786] Out of memory: Killed process 30171 (bash) total-vm:2473468kB, anon-rss:1662772kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:4452kB oom\_score\_adj:0  
[ 13336.832440] oom\_reaper: reaped process 30171 (bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB**

* 1. **report2.log:**

**<…>**

**13000000**

**14000000**

* 1. **report.log:**

**<…>**

**27000000**

**28000000**

1. Результаты замеров c использованием **follower.bash** из **run1.bash**:
2. Выводы:

Всё аналогично первому этапу, но т. к. процесса два, память заполняется равными темпами двумя процессами, как только два процесса заполняют имеющуюся оперативную память, начинает происходить страничный обмен – процесс подкачки забирает часть ресурсов, уменьшается потребление памяти, в какой-то момент один из процессов превышает весь доступны объём памяти и аварийно завершается, для второго процесса высвобождается память, при этом часть его данных хранится в подкачке, а другая часть в оперативной памяти (освободившаяся память от второго процесса аналогично) – сначала опять заполняется оперативная память, а потом снова запускается процесс подкачки, пока вся память не будет использовано – далее второй процесс завершается аварийно.y

**Эксперимент №2**

Задача – определить граничные значения потребления памяти, обеспечивающие безаварийную работу для регулярных процессов, запускающийся с заданной интенсивностью.

1. С предложенными параметрами в системном журнале отсутствуют отметки об аварийных завершениях.
2. Результаты замеров c использованием **follower.bash** из **run3.bash (N = 3.000.000)**:

1. Бинарным поиском исследуем, когда все 30 процессов будут завершаться штатно. Результаты замеров c использованием follower.bash из run3.bash (N = 1.630.000):