**Операционные системы**

Лабораторная работа 5

Осиков Александр

**Данные конфигурации**

* Общий объем оперативной памяти: 12987080
* Объем раздела подкачки: 4194304
* Размер страницы виртуальной памяти: 4096
* Объем свободной физической памяти в ненагруженной системе: 12814240
* Объем свободного пространства в разделе подкачки в ненагруженной системе: 4123952

**Эксперимент 1.**

**Первый этап**

* Память заполнялась равномерно. Процесс сразу же попал на ***первую строчку top*** и оставался там до его завершения. Сначала использовалась физическая память, а когда она закончилась, стал использоваться раздел подкачки.
* При заполнении памяти на ***97,9%*** процесс был завершён *xmalloc* c данной ошибкой:

*./mem.bash: xmalloc: cannot allocate 2 bytes*

*[1]+ Terminated ./mem.bash*

* Последний размер массива, записанный в report.log - ***216000000***

**Второй этап**

* Во втором этапе наблюдается похожая ситуация, как и в первом.
* Процессы заняли ***первую и вторую строчку top***. Процессы равномерно использовали память, пока в сумме не заняли ***97.8%*** (***48.4%*** и ***49.4%***).
* Потом были завершены xmalloc с ошибкой:

*./mem2.bash: xmalloc: cannot allocate 3 bytes*

*./mem.bash: xmalloc: cannot allocate 2 bytes*

* Размер массива в *report.log* – ***107000000***
* Размер массива в *report2.log* – ***108000000***
* Так как из-за *xmalloc* процессы завершили себя сами, то никакой информации об этом в *dmesg* нет.

**Эксперимент 2**

При *k = 10, n = 21600000* процессы запустились и **полностью отработали**. В пиковые моменты, когда одновременно было запущено 10 процессов использовалось более *90%* ОЗУ. 10 одновременно работающих процессов занимали приблизительно столько же памяти, сколько 1 в первом эксперименте.

При *k = 30, n = n = 21600000* большинство процессов **завершились** из-за нехватки памяти с помощью xmalloc. Случалось это в тот момент, когда память уже была загружена на 99%, а процесс запрашивал ещё больше памяти.

При *k = 30*, максимальное *n* при котором все процессы корректно завершают свою работу – **13125000**, это значение было получено двоичным поиском, где в качестве верхней границы было *21600000*, известное из прошлого пункта, а нижняя граница – максимальное n из первого эксперимента / *30* = *8700000*. При этом физическая память заполнялась полностью, а раздел подкачки на *98%*.