Отчет

Первый эксперимент:

Первый этап:

1. Две записи о скрипте в системном журнале:

Изображение выглядит как текст

Автоматически созданное описание

1. Последние значение в файле reports.log – 3400000
2. Данные, полученные во время наблюдения с помощью команды top:

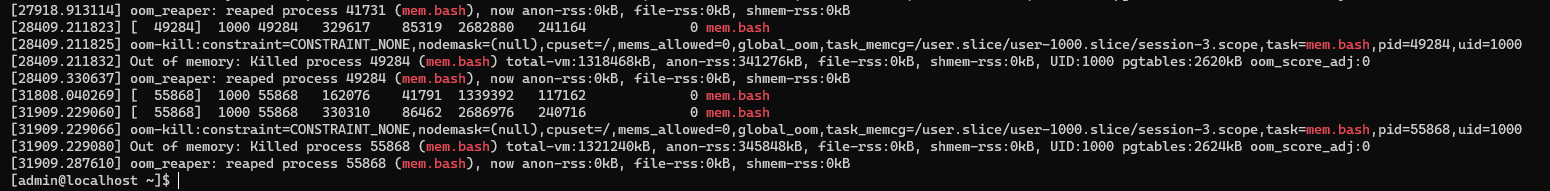
Изображение выглядит как текст

Автоматически созданное описание

1. График зависимости времени от памяти:

Второй этап:

1. Добавляем ещё один скрипт
2. Значение последней строки в report2.log: 8500000; report.log: 16700000





|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TIME | MEM | VIRT | RES | SHR | CPU | FREE | SWAP |
| 0:01.87 | 3,9 | 28300 | 18660 | 2940 | 99,5 | 198,10 | 944,80 |
| 0:15.36 | 26,8 | 137068 | 127428 | 2940 | 96 | 24,60 | 909,40 |
| 0:24.67 | 33,2 | 211120 | 158204 | 2316 | 76,5 | 9,90 | 853,70 |
| 0:28.17 | 34,4 | 238444 | 163912 | 2316 | 90,2 | 10,00 | 799,30 |
| 0:38.24 | 34,9 | 318172 | 166348 | 2180 | 79,7 | 7,00 | 480,00 |
| 0:55.50 | 34,6 | 452152 | 164964 | 1508 | 91,2 | 10,30 | 394,90 |
| 1:01.78 | 35,2 | 501784 | 167512 | 1508 | 55,6 | 11,70 | 230,20 |
| 1:10.00 | 33,5 | 565936 | 159548 | 1508 | 86,4 | 14,20 | 167,80 |
| 1:13.30 | 33,2 | 591676 | 158096 | 1508 | 89,3 | 8,10 | 131,90 |
| 1:18.87 | 33,5 | 634840 | 159732 | 1504 | 67,4 | 7,20 | 471,20 |

При низкой величине физической памяти (7-10 Мб) происходит переход на файлы подкачки. Как только их значение станет критично, то произойдет аварийное завершение программы.

Если запустить два скрипта одновременно, то их графика можно заметить, что один из них продолжил работу, это обусловлено тем, что во время аварийного завершения первого скрипта у машины восстанавливается физическая память, далее идет сценарий как из первого этапа.

Второй эксперимент:

Изменил mem и добавил условие на прекращение скрипта и написал обработчик

При K = 10, N = 3200000 программа успешно завершила свое выполнение

Изображение выглядит как текст

Автоматически созданное описание

При K = 30, N = 3400000 программа аварийно завершила свое выполнение

Изображение выглядит как текст

Автоматически созданное описание

При K = 30, N = 2000000 программа аварийно завершила свое выполнение

Изображение выглядит как текст

Автоматически созданное описание

При K = 30, N = 1800000 программа успешно завершила свое выполнение

Изображение выглядит как текст

Автоматически созданное описание

При K = 30, N = 1900000 программа успешно завершила свое выполнение

Изображение выглядит как текст

Автоматически созданное описание

Вывод: доказал, что во время работы с оперативной памятью данные заполняют физическую память, пока не дойдет до критического значения. Далее ОС начинает заполнять файлы подкачки. В случае, когда файлов подкачки будет не хватать процесс аварийно завершится.