**ФПИиКТ**

**Отчет по дисциплине ОС**

**Лабораторной работе №1**

**Вариант №334886**

Ибадуллаев Алибаба Эльбрус оглы

Группа: P33131

Преподаватель:  [Осипов Святослав Владимирович](https://my.itmo.ru/persons/224794)

Санкт-Петербург

2022г.

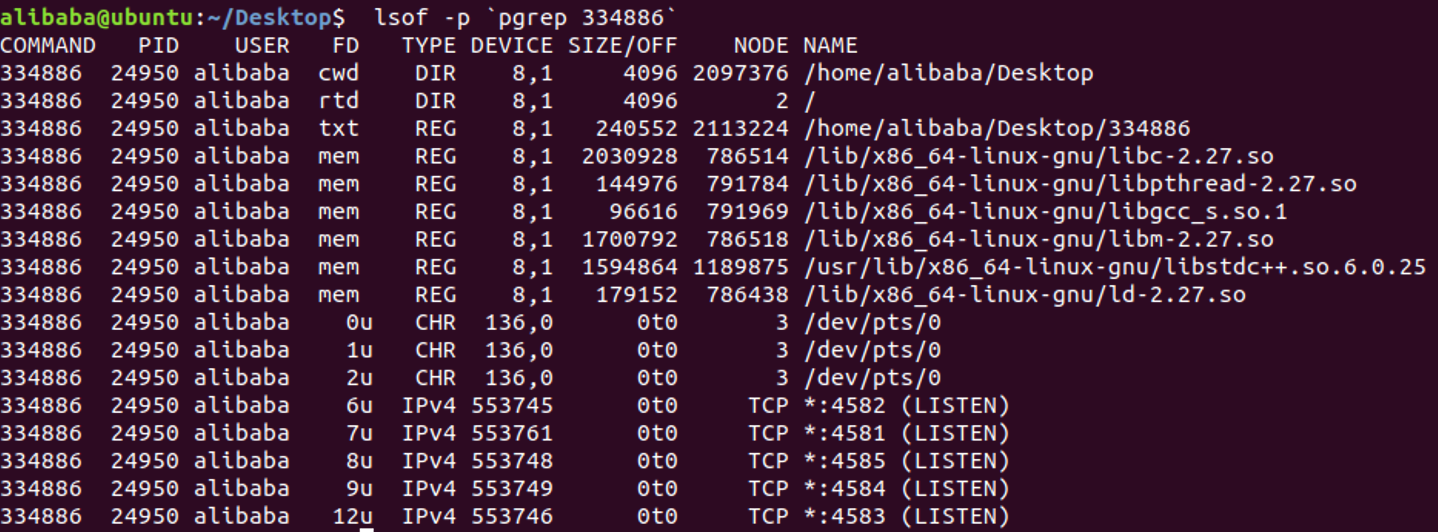
1. Количество потоков создаваемое программой

**ls /proc/`pgrep 334886`/task | wc -l**



1. Список файлов и сетевых соединений с которыми работает программа

**lsof -p `pgrep 334886`**



**sudo strace -fe trace=file -p `pgrep 334886`**

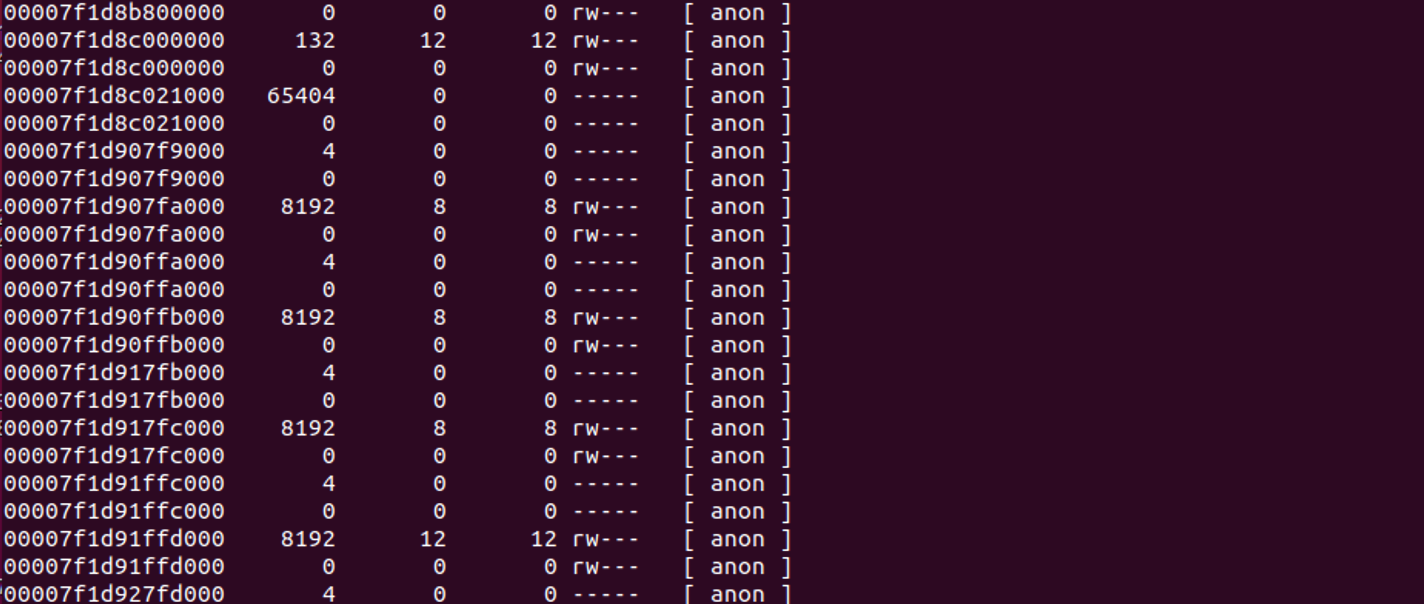
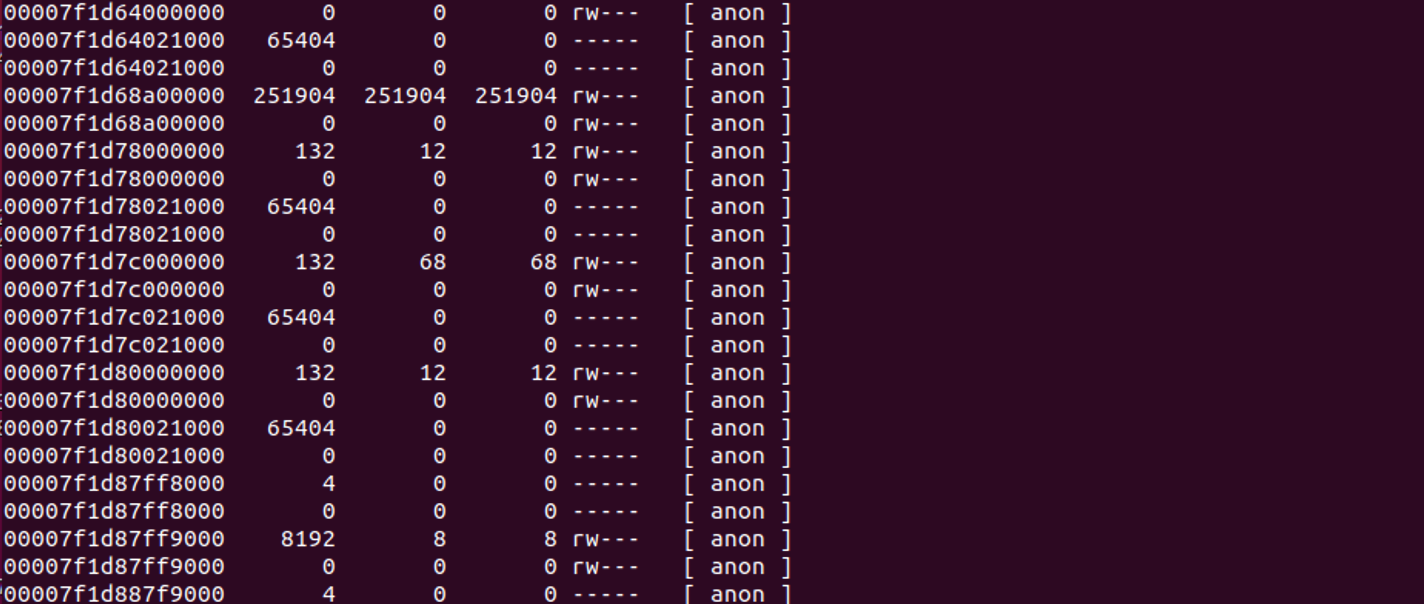
Изображение выглядит как текст

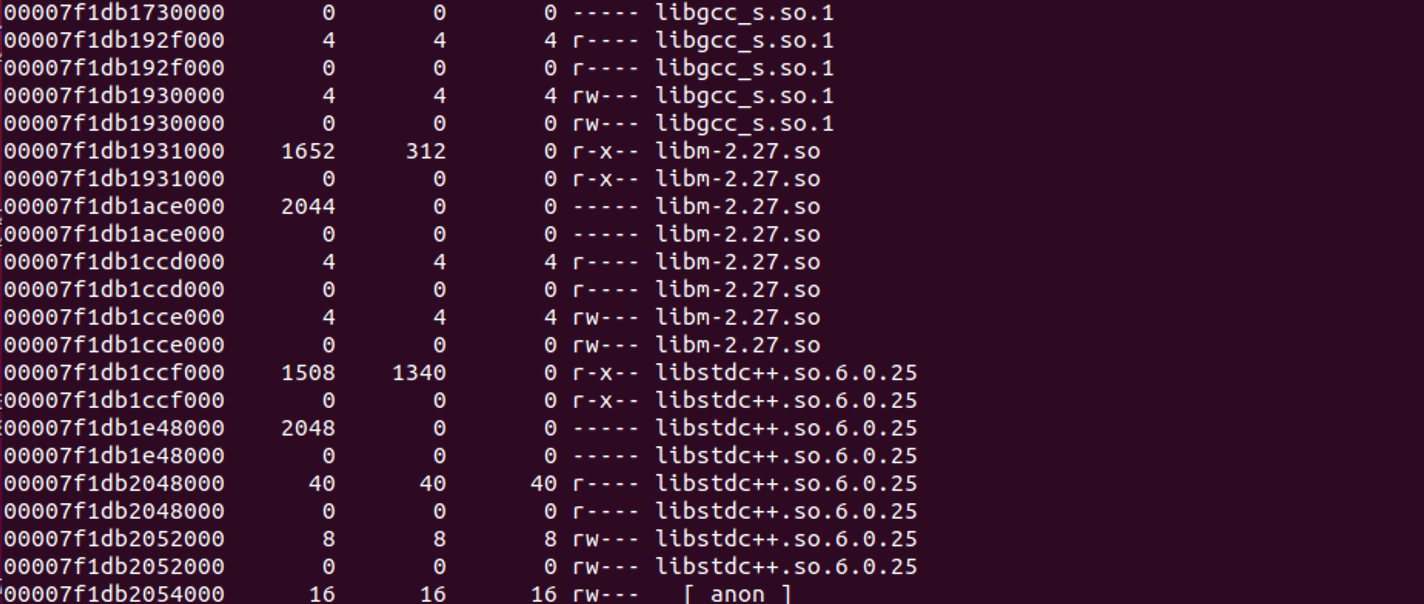
Автоматически созданное описание

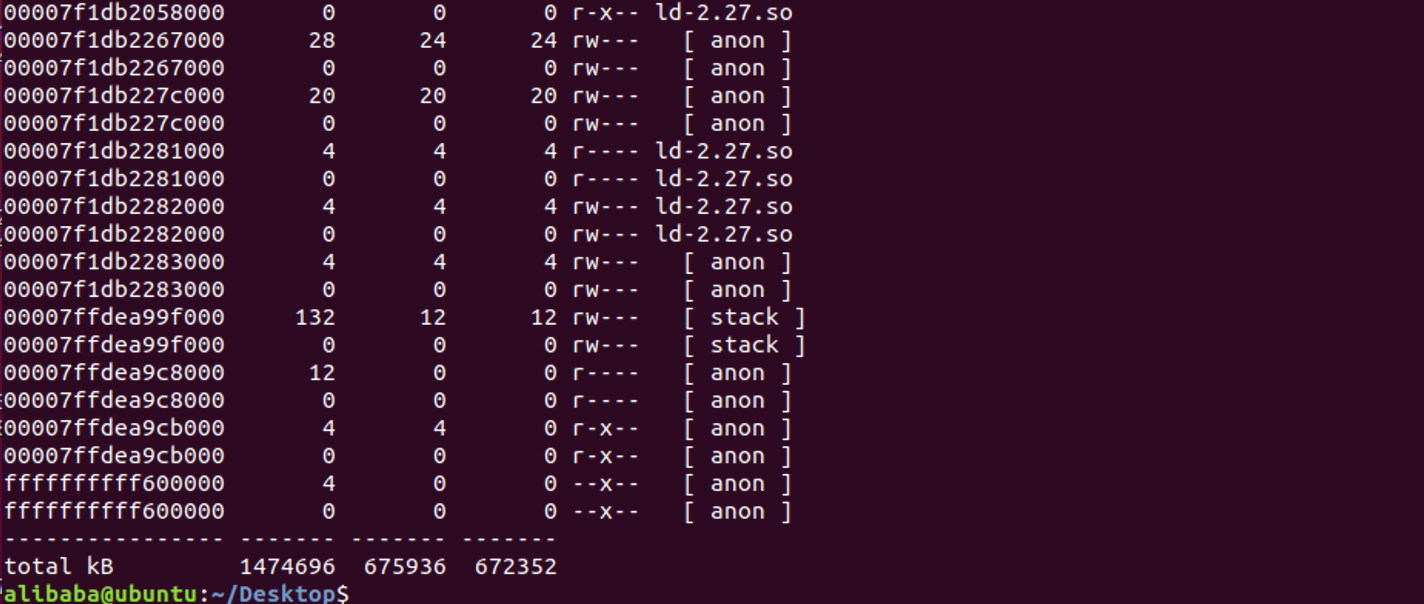
1. Карта памяти процесса

**ls -la /proc/`pgrep 334886`/maps or pmap -xx `pgrep 334886`**

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

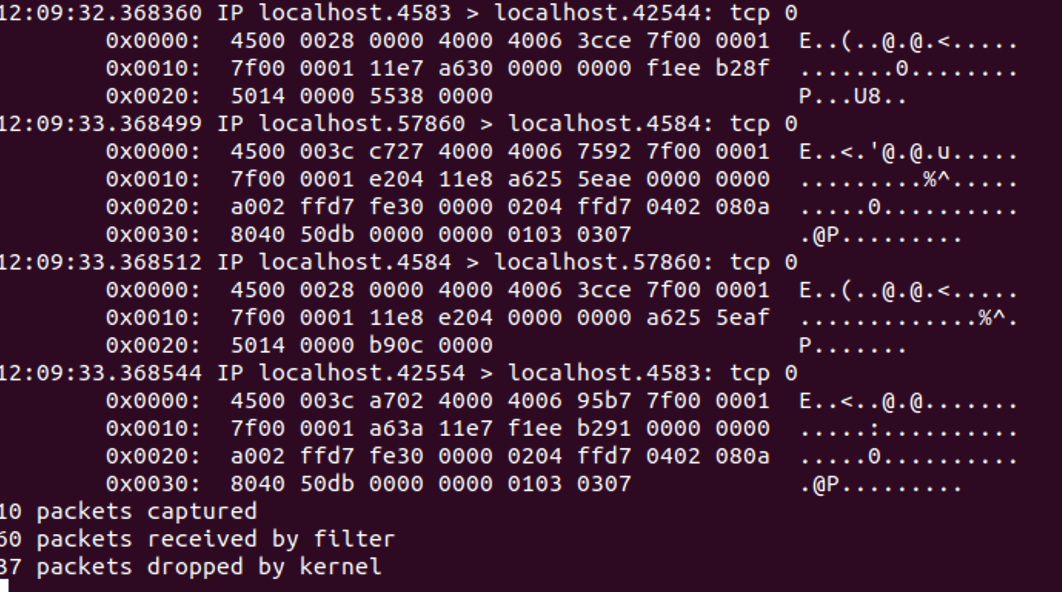
Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

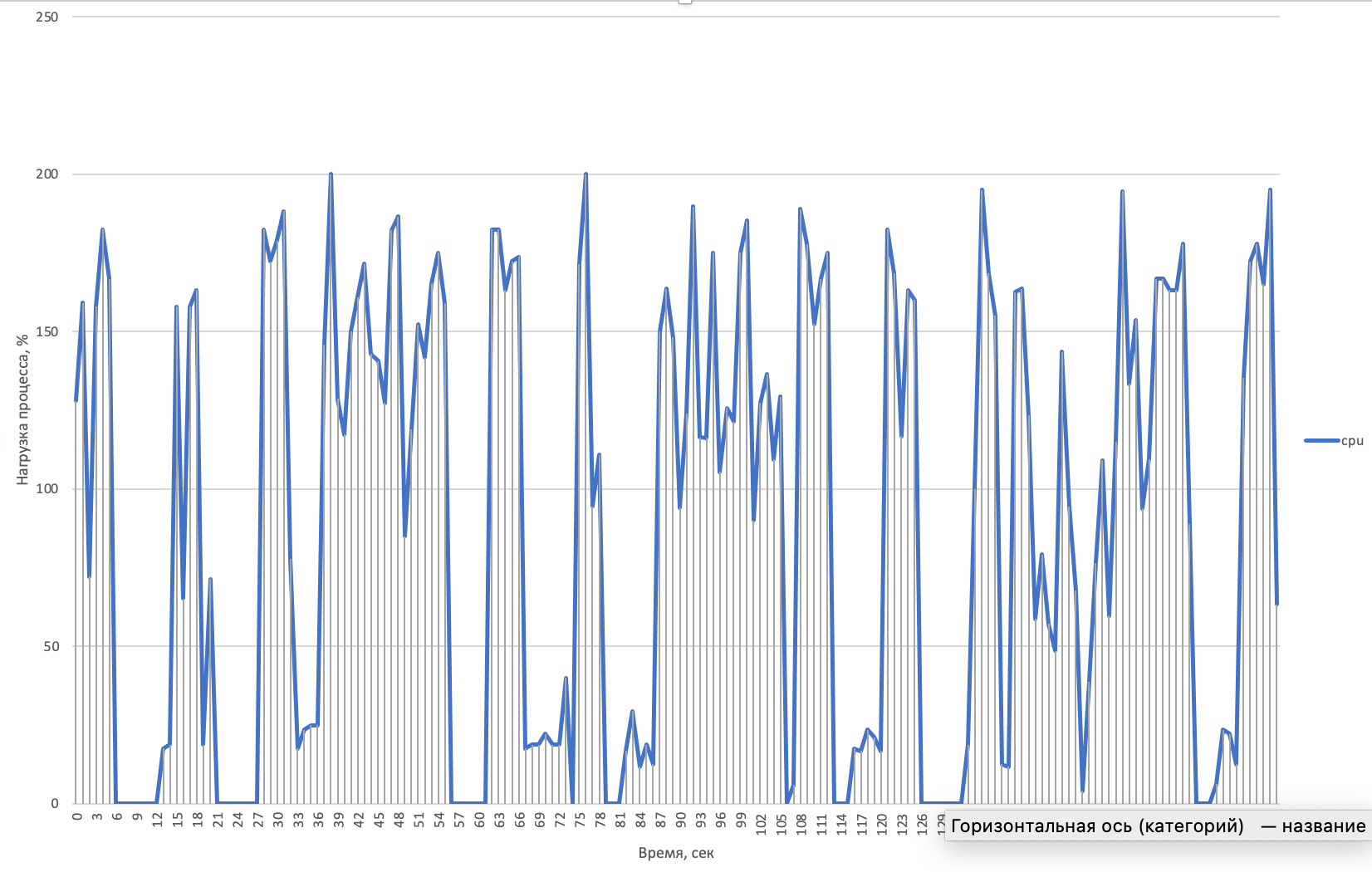
1. Содержимое передаваемых по сети данных;

**sudo tcpdump -i any -X -c10 -q**

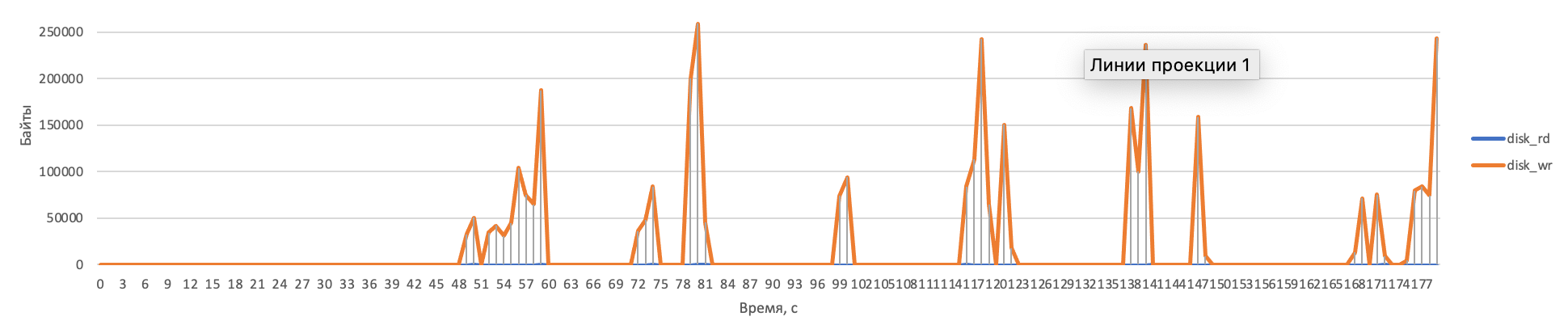
Изображение выглядит как текст

Автоматически созданное описание

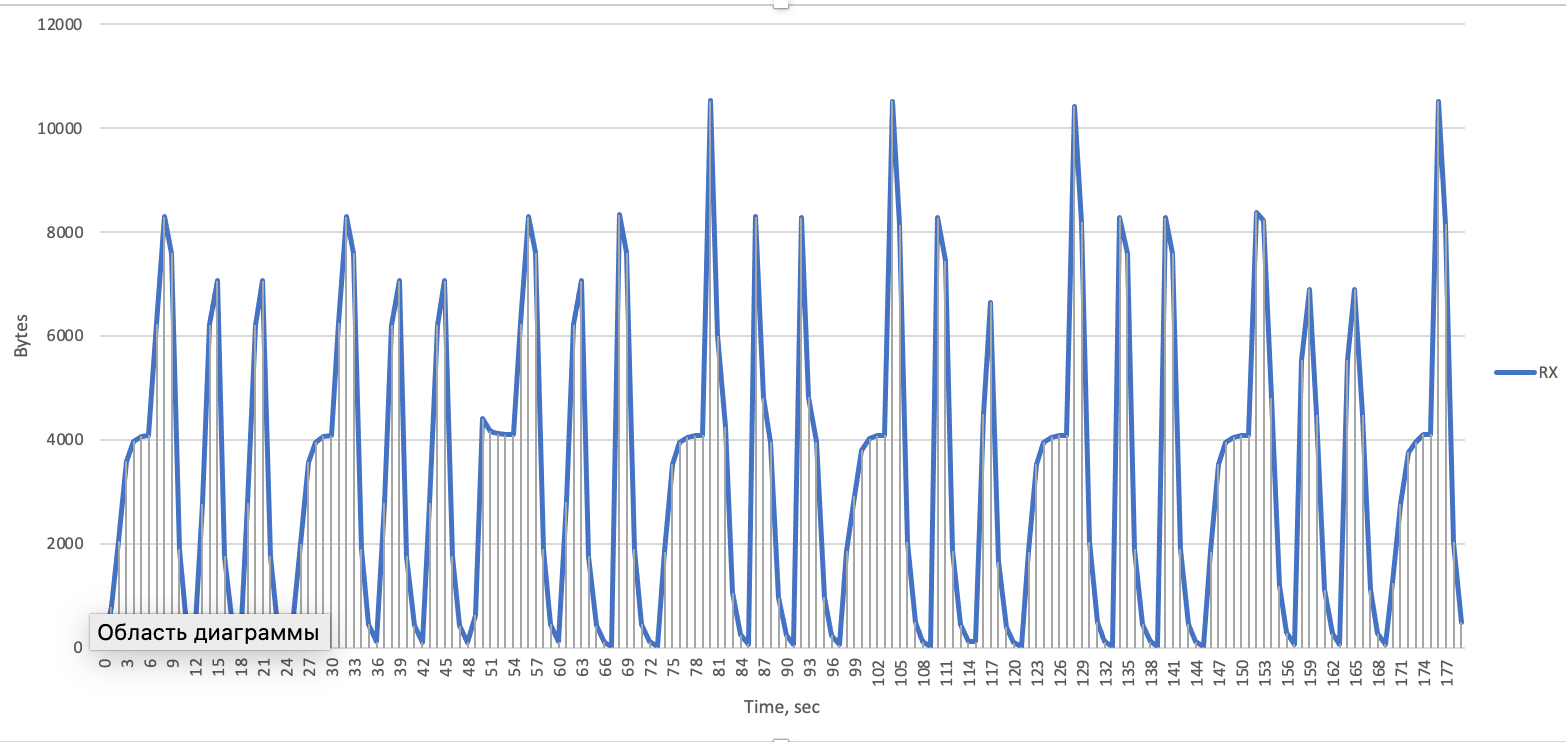
Потребления программой cpu



Нагрузки генерируемой программой на подсистему ввода-вывода;



Нагрузки генерируемой программой на сетевую подсистему.



Смены состояния исполнения потоков

Изображение выглядит как стол

Автоматически созданное описание

Скрипт для выполнения задания про графики.

1. #! /bin/bash
2. VARIANT\_NUMBER="334886"
3. PORTS\_OF\_PROC=$(sudo lsof -i -a -p `pgrep ${VARIANT\_NUMBER} ` -e /run/user/1000$
4. PORTS\_OF\_PROC=${PORTS\_OF\_PROC::-1}
5. CPU\_ITER=180
6. CPU\_TIME\_ITER=0
7. IO\_ITER=180
8. IO\_TIME\_ITER=0
9. NET\_ITER=180
10. NET\_TIME\_ITER=0
11. THREADS\_ITER=21
12. SLP\_TIME=1
13. if pgrep -x ${VARIANT\_NUMBER} > /dev/null
14. then
15. echo $PORTS\_OF\_PROC
16. PORTS\_FOR\_DUMP=$(sed 's/ / or port /g' <<< $PORTS\_OF\_PROC)
17. if [ -z "$PORTS\_OF\_PROC" ]
18. then
19. sudo tcpdump -i any -X -c10 -q
20. else
21. sudo tcpdump -i any port $PORTS\_FOR\_DUMP -X -c10 -q
22. fi
23. # CPU usage
24. val=$(( $SLP\_TIME \* $CPU\_ITER))
25. echo "CPU USAGE TEST STARTED! IT WILL END IN $val SEC"
26. for i in `seq 1 $CPU\_ITER`
27. do
28. echo $(echo "$CPU\_TIME\_ITER";
29. top -b -n 1 |
30. grep 334886 |
31. grep -oP '(?:\s)[0-9]\*\.[0-9]\*'
32. ) | tr ' ' ',' >> cpu.csv;
33. sleep $SLP\_TIME;
34. ((CPU\_TIME\_ITER = CPU\_TIME\_ITER + 1))
35. Done
36. # IO usage
37. val=$(( $SLP\_TIME \* $IO\_ITER))
38. echo "IO USAGE TEST STARTED! IT WILL END IN $val SEC"
39. sudo pidstat -d -p `pgrep 334886` 1 $IO\_ITER > temp.csv
40. sed -i '1,3 d' temp.csv
41. sed -i '$ d' temp.csv
42. echo "date,disk\_rd,disk\_wr"> io.csv;
43. awk -v time\_iter="$IO\_TIME\_ITER" -v
44. OFS=, '{print time\_iter++ , $5, $6}' temp.csv >> io.csv
45. rm temp.csv
46. ((IO\_TIME\_ITER = IO\_TIME\_ITER + 1))
47. # NET usage
48. val=$(( $SLP\_TIME \* $NET\_ITER))
49. echo "NET USAGE TEST STARTED! IT WILL END IN $val SEC"
50. echo "time,RX,TX" > net.csv
51. timeout $NET\_ITER bmon -p lo -o ascii -b > in.txt
52. awk 'NR % 2 == 0' in.txt > out.txt
53. rm in.txt
54. awk '{ system(" echo $(python pyth.py "$2") >> in.txt "); }' out.txt
55. awk -v time\_iter="$NET\_TIME\_ITER" -v
56. OFS=, '{print time\_iter++,$0}' in.txt >> net.csv
57. rm in.txt
58. rm out.txt
59. # THREADS usage
60. val=$(( $SLP\_TIME \* $THREADS\_ITER))
61. echo "THREADS USAGE TEST STARTED! IT WILL END IN $val SEC"
62. rm proc.csv
63. ps -T -p `pgrep 334886` S | awk '{print $2}' >> proc.csv
64. for ((i=0; i< $THREADS\_ITER; i++));
65. do
66. ps -T -p `pgrep 334886` S | awk '{print $4}' >> in.csv
67. python proc.py
68. rm in.csv
69. sleep 0.5
70. done
71. else
72. echo "Please check that your program running!"
73. fi

Вывод: я познакомился с утилитами для мониторинга ОС Linux. Также применил на практике команды необходимые для реализации гравиков и как следствие написал скрипты для вывода их в файл. Ознакомился с понятиями виртуальной памяти, устройства процесса, файловой системы, устройства ввода вывода и применил их на практике.