НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет ПИиКТ

Операционные системы

Лабораторная работа № 1

Вариант 60e030

Выполнил студент

Венщиков Марат

Группа № P33302

Санкт-Петербург

2021

**Задание:**

Основная цель лабораторной работы - знакомство с системными инструментами анализа производительности и поведения программ. Для этого предлагается для выданной по варианту программы выяснить следующую информацию:

1. Количество потоков создаваемое программой;
2. Список файлов и сетевых соединений с которыми работает программа
3. Карту памяти процесса;
4. Содержимое передаваемых по сети данных;
5. Построить графики:

* Потребления программой cpu;
* Нагрузки генерируемой программой на подсистему ввода-вывода;
* Нагрузки генерируемой программой на сетевую подсистему.
* Смены состояния исполнения потоков;

Содержание отчета:

1. Описание шагов выполненных для сбора информации (включая исходные тексты всех использованных скриптов и вспомогательных программ);
2. Полученные графики;
3. Выводы по работе.

**Отчет:**

1. **Количество потоков**

Было получено с помощью команды ps

ps -T -p $PID | tail -n +2

Выдала список всех потоков процесса (35).

1. **Список файлов и сетевых соединений**

lsof -p $PID

1. **Карта памяти**

cat /proc/$PID/maps

1. **Содержимое передаваемых по сети данных**

Было получено двумя способами:

strace -p $PID -f -e trace=network -t

Лог системных вызовов связанных с сетью.

tcpdump -i any -A port [port]

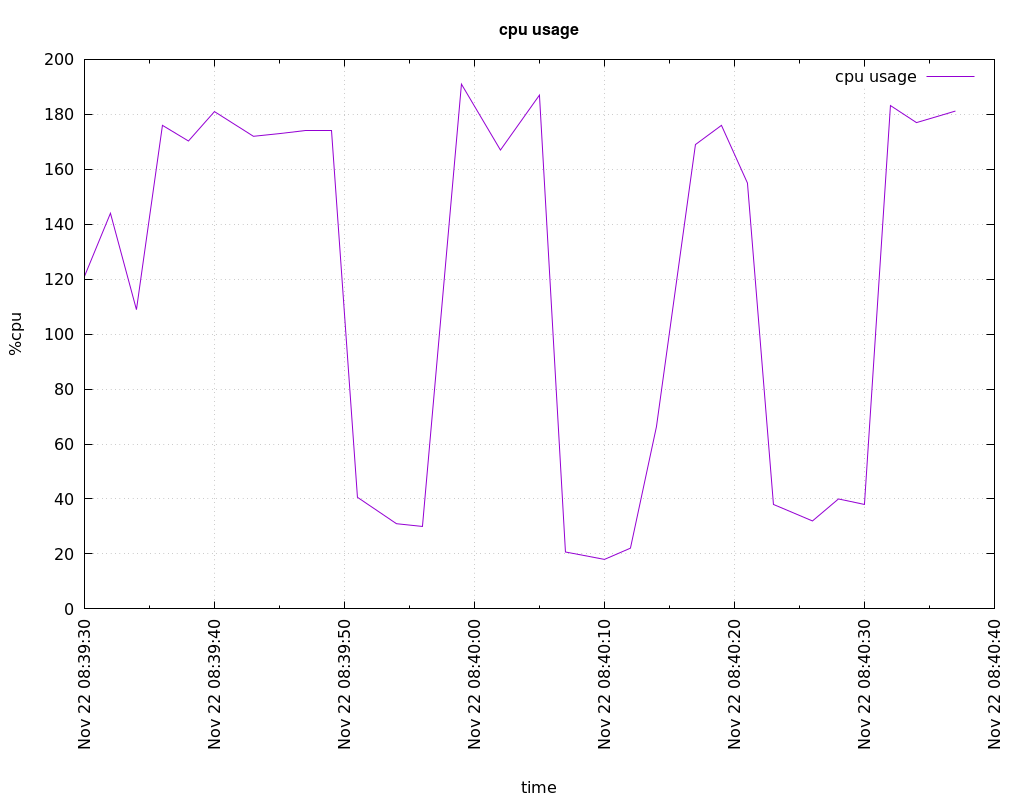
Список пакетов, отправленных/полученных на указанные порты.

1. **Потребление программой CPU**

timestamp=$(date +"%b %d %H:%M:%S")

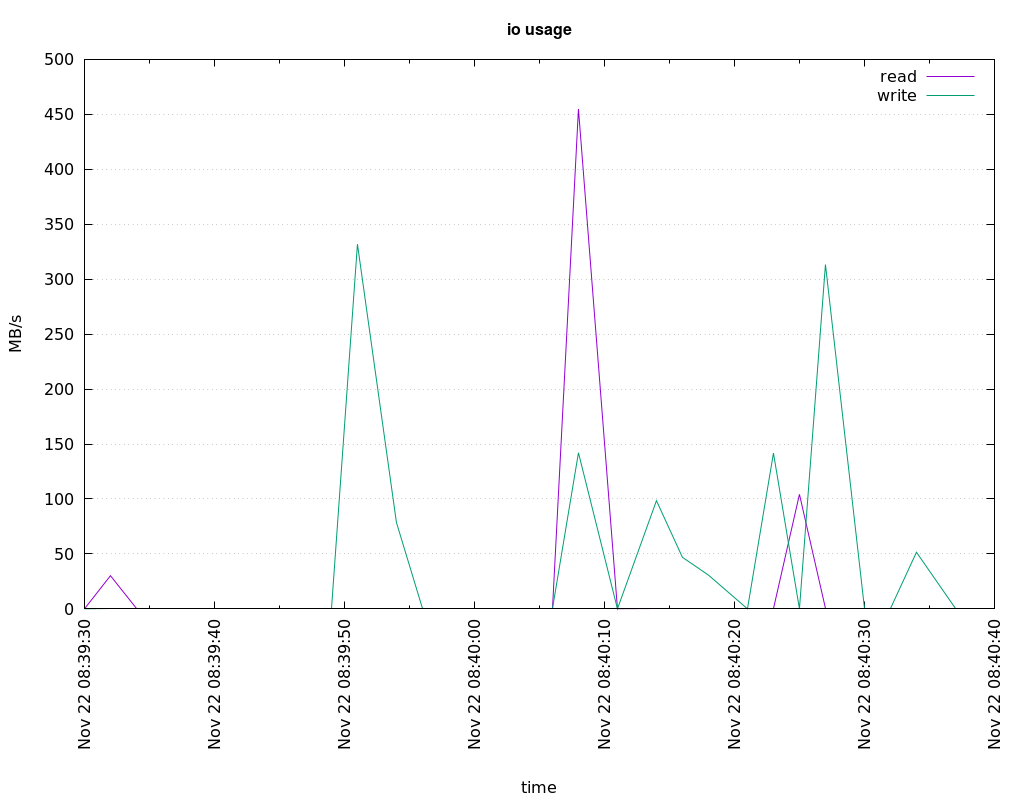
cpu=$(top -b -n 2 -d 1 -p $1 | tail -1 | awk '{print $9}')

echo "$timestamp;$cpu" >> $csv\_filename



1. **Нагрузка генерируемая программой на подсистему ввода-вывода**

io=$(iotop -o -qqq -b -n 1 -P -k)

io=$(echo ${io} | grep "${pid}" ) 

1. **Нагрузка генерируемая программой на сетевую подсистему**

global r, s

probe netdev.transmit

{if(pid() == target())

s <<< length}

probe netdev.receive

{if(pid() == target())

r <<< length}

function print\_activity()

{printf("%10s;%5d;%5d;%5d;%5d\n",

substr(ctime(gettimeofday\_s()), 4, 21),

@count(s), @count(r),

@count(s) ? @sum(s) : 0,

@count(r) ? @sum(r) : 0)

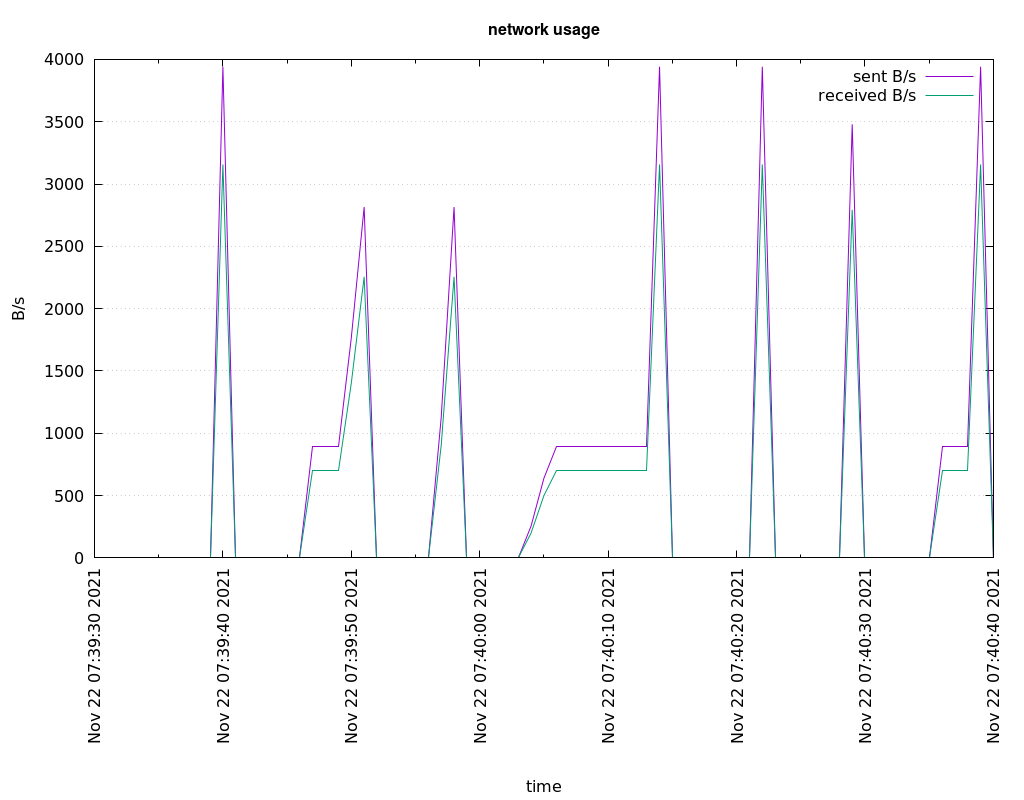
delete r

delete s

if(pid2execname(target()) == "") {exit()}

}

probe timer.ms(1000), end, error{print\_activity()}



1. **Смена состояния исполнения потоков**

timestamp=$(date +"%b %d %H:%M:%S")

top=$(top -H -p $pid -b -n 1 | tail -n +8 | awk '{print $1, $8"\n"}')

R=$(echo "$top" | grep "R" | wc -l)

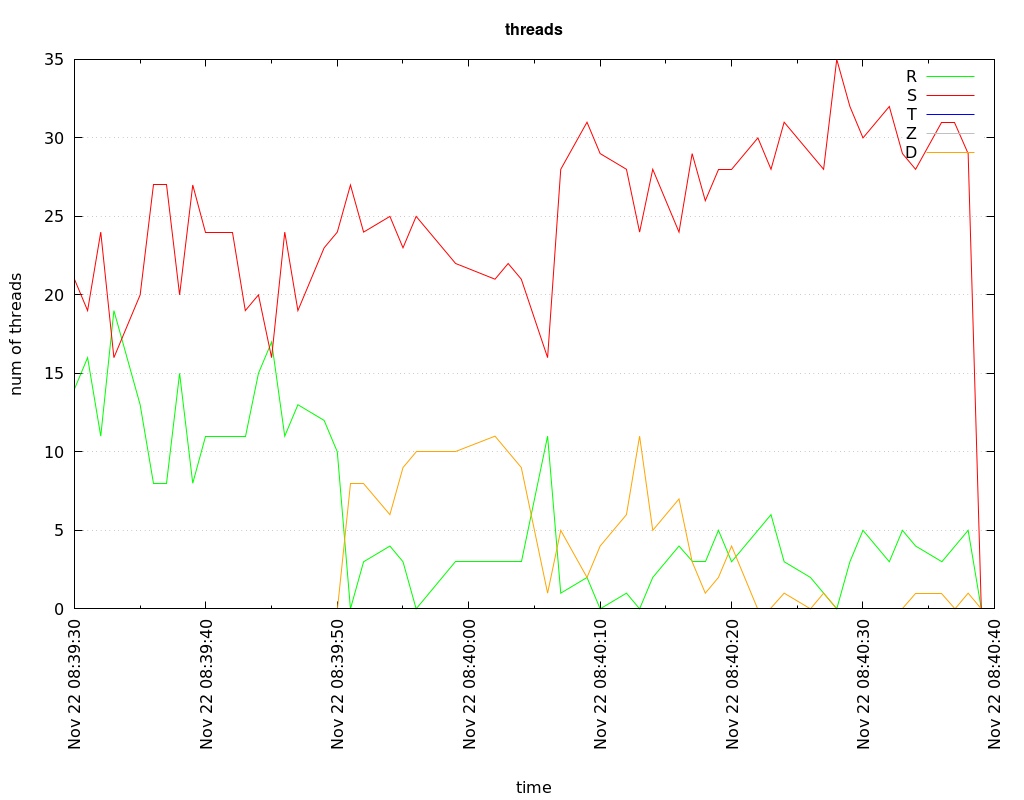
S=$(echo "$top" | grep "S" | wc -l)

T=$(echo "$top" | grep "T" | wc -l)

Z=$(echo "$top" | grep "Z" | wc -l)

D=$(echo "$top" | grep "D" | wc -l)

echo "$timestamp;$R;$S;$T;$Z;$D" >> $csv\_filename



**Исходные коды**

https://github.com/yellowblacksnek/oslabs

**Вывод**

Попользовался системными инструментами анализа производительности и поведения программ, узнал механизмы оценки загруженности процессора, системы ввода-вывода и сетевой подсистемы. Научился извлекать полезную информацию из лога системных вызовов. Ознакомился с systemtap.