

Предмет	Управление и автоматизация баз данных
Курс	4
Семестр	1
Работа	Lab09 rrdtool
Дата	14.10.2021
Группа	484
Фамилия	Серебряков
Имя	Максим
Отчество	Андреевич

1. Shell-сценарий для создания новой базы данных RRD с целью хранения времени.

```
File Edit View Search Terminal Help
GNU nano 3.2 log-init.sh

DB=mr6363_1
start=$(date +%s)
echo $start>log-start.txt
interval=15
name=my_ser
type=GAUGE
min=5
max=1500
step=10
vals=5
rows=1000
func=AVERAGE
cap30=$(expr 30 \* 60)
cap60=$(expr 60 \* 60)
stv=$(expr $step \* $vals)
rows30=$(expr $cap30 / $stv)
rows60=$(expr $cap60 / $stv)
rrdtool create log_data.rrd \
--start $start \
--step $step \
DS:$name:$type:$interval:$min:$max \
RRA:$func:0.5:$vals:$rows \
RRA:$func:0.5:$vals:$rows30 \
RRA:$func:0.5:$vals:$rows60
```

log-init.sh (с вычисляемым vals)

2. Сценарий на языке Lua, который считывает в качестве входных данных - вывод команды ping, находит среднее время отклика с использованием регулярных выражений.

```
File Edit View Search Terminal Help
GNU nano 3.2 ping.lua

local s = io.read('*a')

local _, _, x = s:find('%d+.%d+/(%d+.%d+)')
io.write(x)
```

Ping.lua

3. Shell-сценарий для автоматического обновления базы данных актуальными значениями времени отклика.

```
File Edit View Search Terminal Help
GNU nano 3.2 log-append.sh

o host=yandex.ru
  host2=rbc.ru
o host3=bbc.com
  while :
o do
  now=$(date +%s)
o val=$(ping -c 2 -q -n $host | lua ping.lua)
  rrdtool update log_data.rrd $now:$val
o echo "added $val at $now"
  now=$(date +%s)
o val2=$(ping -c 2 -q -n $host2 | lua ping.lua)
  rrdtool update log_graph2.rrd $now:$val2
o echo "added $val2 at $now"
  now=$(date +%s)
o tval3=$(ping -c 2 -q -n $host3 | lua ping.lua)
  rrdtool update log_graph3.rrd $now:$val3
  echo "added $val3 at $now"
  sleep 10
V done
```

Log-append.sh (с тремя хостами)

4. Проверьте состояние базы данных командой rrdinfo

```
user@Serebriacov:~/gitup/rrdtoolGraph$ rrdinfo log_data.rrd
filename = "log_data.rrd"
rrd_version = "0003"
step = 10
last update = 1634194164
header_size = 1000
ds[my_ser].index = 0
ds[my_ser].type = "GAUGE"
ds[my_ser].minimal_heartbeat = 15
ds[my_ser].min = 5.0000000000e+00
ds[my_ser].max = 1.5000000000e+03
ds[my_ser].last_ds = "15.281"
ds[my_ser].value = 6.1124000000e+01
ds[my_ser].unknown_sec = 0
rra[0].cf = "AVERAGE"
rra[0].rows = 1000
rra[0].cur_row = 438
rra[0].pdp_per_row = 5
rra[0].xff = 5.0000000000e-01
rra[0].cdp_prep[0].value = 1.5281000000e+01
rra[0].cdp_prep[0].unknown_datapoints = 0
rra[1].cf = "AVERAGE"
rra[1].rows = 36
rra[1].cur_row = 22
rra[1].pdp_per_row = 5
rra[1].xff = 5.0000000000e-01
rra[1].cdp_prep[0].value = 1.5281000000e+01
rra[1].cdp_prep[0].unknown_datapoints = 0
rra[2].cf = "AVERAGE"
rra[2].rows = 72
rra[2].cur_row = 41
rra[2].pdp_per_row = 5
rra[2].xff = 5.0000000000e-01
rra[2].cdp_prep[0].value = 1.5281000000e+01
```

Rrdinfo

5. Сценарий для выборки данных из rrd файла, используя команду rrdtool fetch

```
GNU nano 3.2 fetch.sh

val=$1
db=log_data
start=$(date +%s)
min=$(expr $1 \* 60)
t0=$(expr $start - $min)
t1=$start
rrdtool fetch $db.rrd AVERAGE -s $t0 -e $t1
rrdtool fetch log_graph2.rrd AVERAGE -s $t0 -e $t1
rrdtool fetch log_graph3.rrd AVERAGE -s $t0 -e $t1
```

Fetch.sh (с тремя rrd)

6. Сценарий для создания графика на основе данных из RRD файла.

```
GNU nano 3.2 graph.sh

db=log_data
name1=my_ser

r1=graph1
r2=graph2
r3=graph3

img=img.png

width=640
height=420

start=$(date +%s)
t0=$(expr $start - 2000)
t1=$start

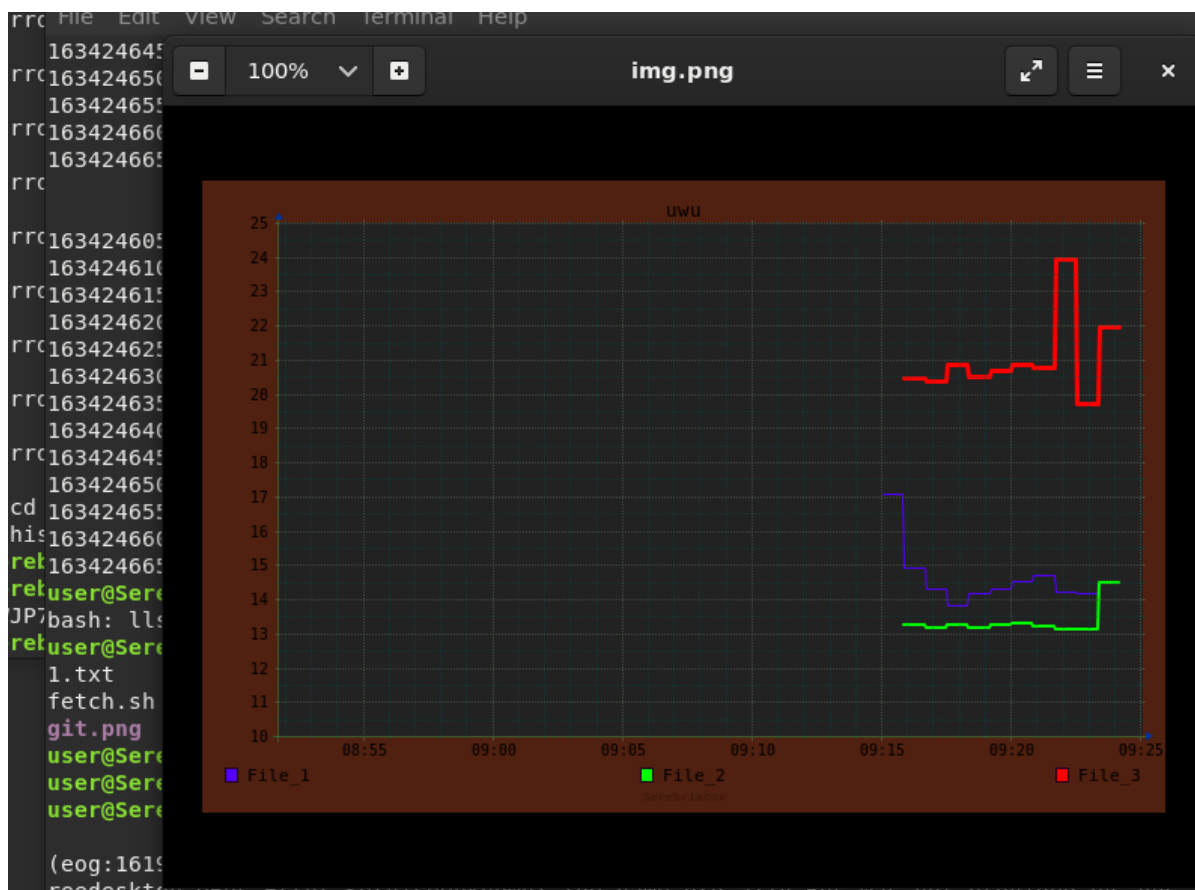
rrdtool graphv $img \
--width=$width \
--height=$height \
--start $t0 --end $t1 \
--full-size-mode \
--disable-rrdtool-tag \
--font-render-mode light \
--border 0 \
--color ARROW#003299 --color AXIS#337733 \
--color FRAME#220022 --color CANVAS#222222 \
--color FONT#ffffff --color BACK#502010 \
--color GRID#005555 --color MGRID#556655
```

```
t1=$start

rrdtool graphv $img \
--width=$width \
--height=$height \
--start $t0 --end $t1 \
--full-size-mode \
--disable-rrdtool-tag \
--font-render-mode light \
--border 0 \
--color ARROW#003299 --color AXIS#337733 \
--color FRAME#220022 --color CANVAS#222222 \
--color FONT#ffffff --color BACK#502010 \
--color GRID#005555 --color MGRID#556655 \
--color FONT#000000 \
--graph-render-mode normal \
--slope-mode \
--title "uwu" \
--watermark "${(cat /etc/hostname)}" \
DEF:$gr1=$db.rrd:$name1:AVERAGE \
DEF:$gr2=log_graph2.rrd:graph2:AVERAGE \
DEF:$gr3=log_graph3.rrd:graph3:AVERAGE \
LINE1:$gr1#5500FF:File_1 \
LINE2:$gr2#00FF00:File_2 \
LINE3:$gr3#FF0000:File_3 \
CDEF:rx=$gr1,$gr2,$gr3,'+', '+'
```

Graph.sh (с watermark и title)

7. Откройте полученное изображение программой eog имя файла



Graph.png (с тремя графиками)

8. Загрузить файлы в публичный GIT репозиторий на github.com вместе с отчетом в PDF.

```
555 git clone https://github.com/MaxSerebrriacov/rrdtoolGraph.git
556 ls
557 git init
558 git clone https://github.com/MaxSerebrriacov/rrdtoolGraph.git
559 ls
560 cd rrdtoolGraph/
561 s
562 ls
563 cd ..
564 cd rrdtoolGraph/
565 git init
566 ls
567 cp ~/gitup/*.sh ~/gitup/rrdtoolGraph/
568 ls
569 cp ~/gitup/*.png ~/gitup/rrdtoolGraph/
570 cp ~/gitup/*.rrd ~/gitup/rrdtoolGraph/
571 ls
572 git add .
573 git commit -m SomeCommit
574 git config --global user.email "paroli0000@yandex.ru"
575 git config --global user.name "MaxSerebrriacov"
576 git add .
577 git commit -m SomeCommit
578 git push
```

Git history

```
user@Serebriacov: ~/gitup/rrdtoolGraph
File Edit View Search Terminal Help
user@Serebriacov:~/gitup/rrdtoolGraph$ ls
1.txt      graph.sh      log_data.rrd  log-initgraph2.sh  README.md
fetch.sh   img.png       log_graph2.rrd log-initgraph3.sh
git.png    log-append.sh log_graph3.rrd log-init.sh
user@Serebriacov:~/gitup/rrdtoolGraph$

user@Serebriacov:~/gitup/rrdtoolGraph$ git add .
user@Serebriacov:~/gitup/rrdtoolGraph$ git commit -m SomeCommit
[main 9166adc] SomeCommit
1 file changed, 1 insertion(+)
create mode 100644 1.txt
user@Serebriacov:~/gitup/rrdtoolGraph$ git push
Username for 'https://github.com': MaxSerebrriacov
Password for 'https://MaxSerebrriacov@github.com':
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 17.60 KiB | 1.35 MiB/s, done.
Total 15 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/MaxSerebrriacov/rrdtoolGraph.git
8d172bc..9166adc main -> main
user@Serebriacov:~/gitup/rrdtoolGraph$
```

Git push