



Компьютерные технологии в научных исследованиях

```
1 #!/bin/bash
2 #INPUT_SAMPLE_LIST=$1
3 cd /Volumes/PhilDrive_EMS/TestDec7/snv_postp
4
11 . paths.txt
12
30
31 echo "Debug level set for $DEBUG_LEVEL"
32 echo "log found in scripts directory"
33
50 cp $HIGH_SNP_OUT ./
51 cp $LOW_SNP_OUT ./
52 cp $GERM_SNP_OUT ./
53 # echo "${SCRIPT_DIR}/run_somatic_mu
54 if [ $DEBUG_LEVEL
55 then
56 echo "INFO: ${SCR
57 `basename ${LOW_S
58 ${D_BAM_FILE} ${G
59
60 fi
61 ${SCRIPT_DIR}run_somatic_mu
62
```

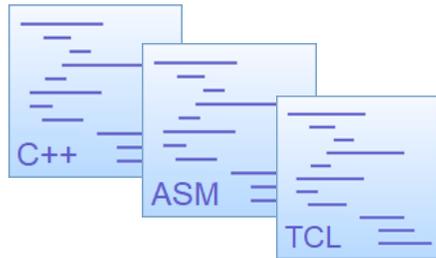


Семинар №3

Текстовые и бинарные данные.
Построение вычислений

Способы представления данных в файлах

Текстовые
(ASCII)



Бинарные
(двоичные)



Запись текстовых данных (на примере языка C++)

```
#include <fstream>
using namespace std;

#define ARRAY_SIZE 10

int main() {
    int mas[ARRAY_SIZE];
    for (int i = 0; i < ARRAY_SIZE; ++i)
        mas[i] = i;

    fstream file_txt("data.txt", ios::out);
    for (int i = 0; i < ARRAY_SIZE; ++i)
        file_txt << mas[i] << " ";
    file_txt.close();

    return 0;
}
```

```
view data.txt - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.txt
0 1 2 3 4 5 6 7 8 9
```

Запись бинарных данных (на примере языка C++)

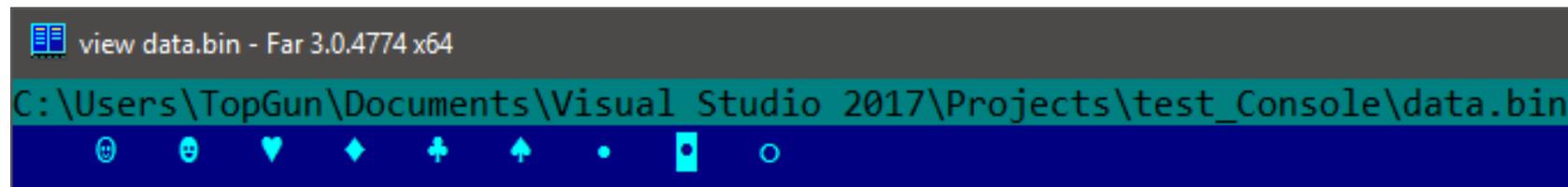
```
#include <fstream>
using namespace std;

#define ARRAY_SIZE 10

int main() {
    int mas[ARRAY_SIZE];
    for (int i = 0; i < ARRAY_SIZE; ++i)
        mas[i] = i;

    fstream file_bin("data.bin", ios::out | ios::binary);
    file_bin.write((char *)&mas, sizeof(int)*ARRAY_SIZE);
    file_bin.close();

    return 0;
}
```



Что именно хранится в бинарном виде?

```
view data.bin - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.bin
```

```
view data.bin - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.bin
0000000000: 00 00 00 00 01 00 00 00 | 02 00 00 00 03 00 00 00 | ☺ ☹ ♥
0000000010: 04 00 00 00 05 00 00 00 | 06 00 00 00 07 00 00 00 | ♦ + ↑ •
0000000020: 08 00 00 00 09 00 00 00 | | █ ○
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☹	♥	♦	♣	♠	●	○							
1	▶	◀					_		↑	↓	→	←		↔	^	▼
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	▯
8	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B	▯	▯	▯		┌	┐	└	┘	┌	┐	└	┘	┌	┐	└	┘
C	┌	┐	└	┘	-	+	└	┘	┌	┐	└	┘	┌	┐	=	└
D	┌	┐	└	┘	┌	┐	└	┘	┌	┐	└	┘	┌	┐	└	┘
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	Ё	ё	≥	≤		J	÷	*	°	·	√	²	³	⁴	⁵	⁶

Преимущество использования бинарных данных (3)

Текстовые данные (файл занимает 20 байт)

```
view data.txt - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.txt
0 1 2 3 4 5 6 7 8 9
```

Бинарные данные (файл занимает 40 байт)

```
view data.bin - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.bin
☹ ☹ ♥ ♦ + ↑ • ☐ ○
```

10 чисел
от 0

Текстовые данные (файл занимает 60 байт)

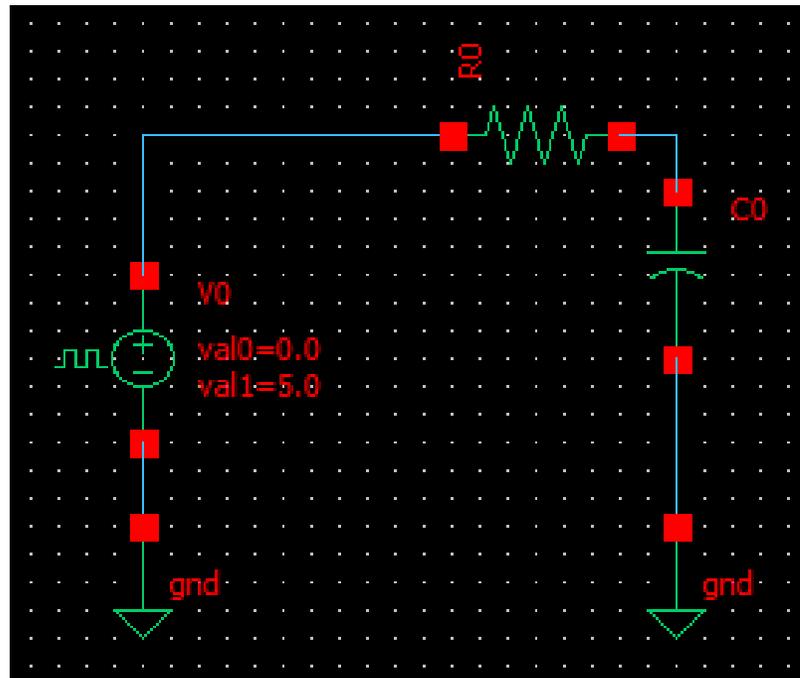
```
view data.txt - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.txt
10000 10001 10002 10003 10004 10005 10006 10007 10008 10009
```

Бинарные данные (файл занимает 40 байт)

```
view data.bin - Far 3.0.4774 x64
C:\Users\TopGun\Documents\Visual Studio 2017\Projects\test_Console\data.bin
←' ←' ↑' !!' '§' §' -' ±' ↑' ↓'
```

10 чисел
от 10000

Текстовые и бинарные данные в САПР (1)

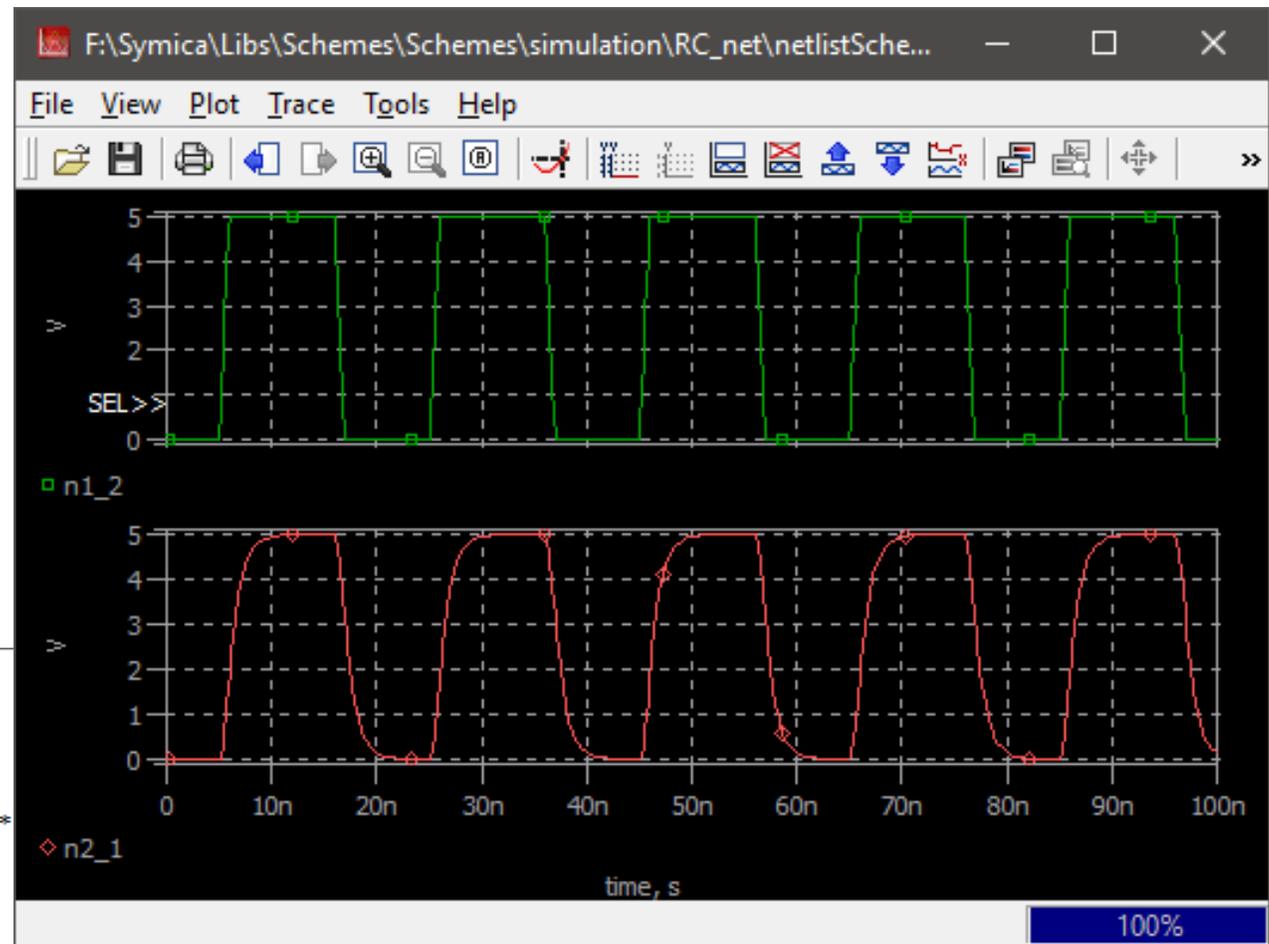


```
*** Title: "F:\Symica\Libs\Schemes\Schemes\simulation\RC_net\netlistSchemes.net"  
*** Generated by: Symica  
*** © 2009-2014 Symica. All rights reserved. (www.symica.com)  
*** Generated for: SymSpice  
*** Generated on: Wed Feb 20 13:49:32 2019  
*****
```

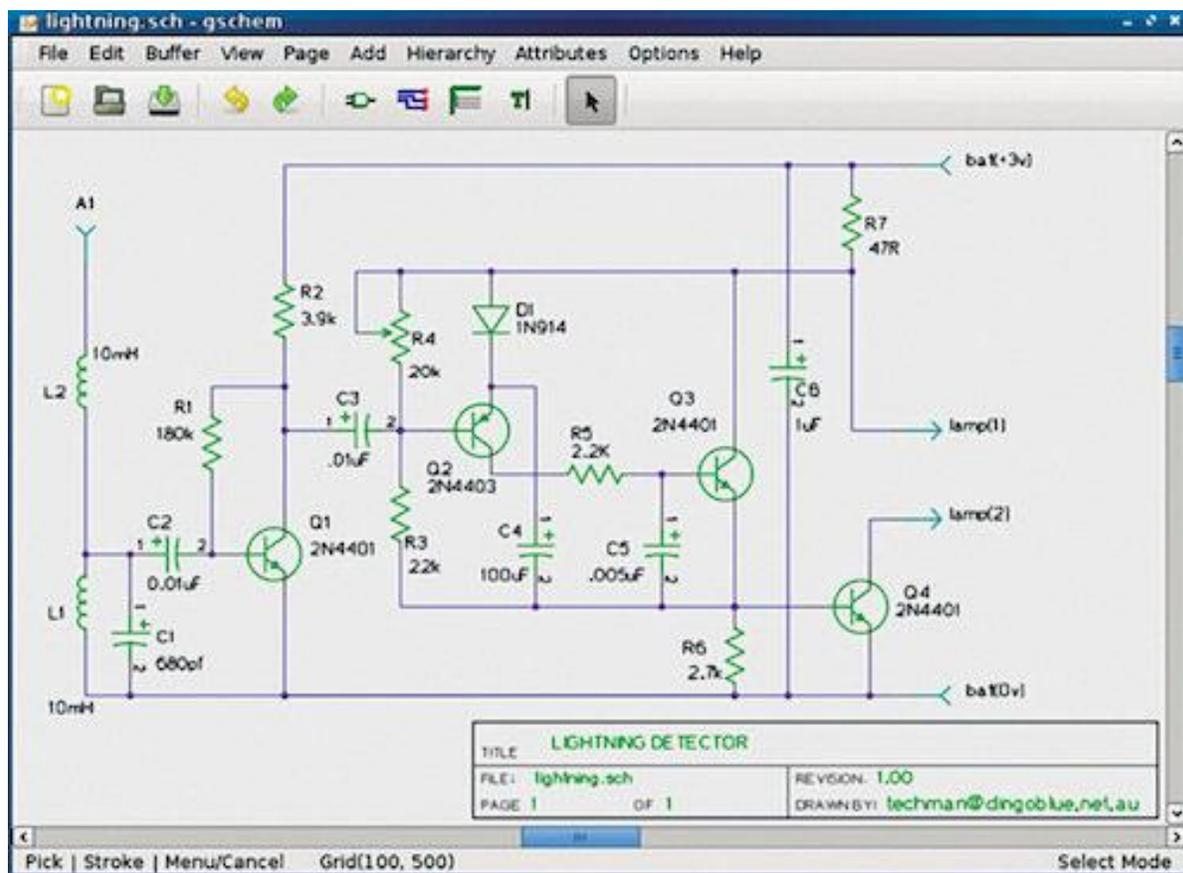
```
simulator lang=local
```

```
parameters
```

```
V0 n1_2 gnd vsource type=pulse val0=0 val1=5 delay=5e-009 period=2e-008 rise=1e-0  
R0 n1_2 n2_1 resistor r=1000  
C0 n2_1 gnd capacitor c=1e-012
```



Текстовые и бинарные данные в САПР (3)



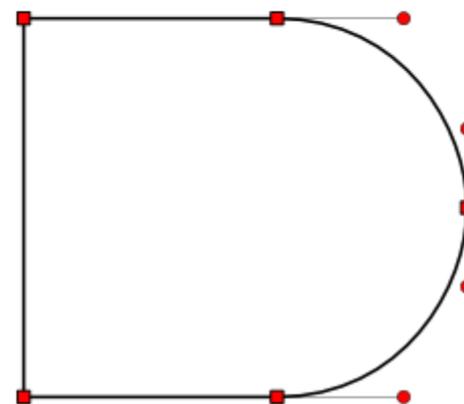
As example, lets draw the outline of an AND gate. The path data is:

```
M 100,100 L 500,100 C 700,100 800,275 800,400  
C 800,525 700,700 500,700 L 100,700 z
```

And a complete schematic:

```
v 20080706 1  
H 3 0 0 0 -1 -1 0 2 20 100 -1 -1 6  
M 100,100  
L 500,100  
C 700,100 800,275 800,400  
C 800,525 700,700 500,700  
L 100,700  
z
```

The resulting path (with control points drawn on to illustrate their positions) is shown here:





Текстовые и бинарные данные в САПР (4)

```
view lab_3.tr0 - Far 3.0.4774 x64
C:\Users\TopGun\Documents
HEADER
"PSFVersion" "1.00"
"simulator" "KSI"
"runtype" "Transient Analy
TYPE
"node" FLOAT DOUBLE PROP(
"key" "node"
)
"branch" FLOAT DOUBLE PRO
"key" "branch"
)
"sweep" FLOAT DOUBLE
SWEEP
"time" "sweep"
TRACE
"group" GROUP 4
"v(1)" "node"
"v(2)" "node"
"v(3)" "node"
"i(d1:1)" "branch"
VALUE
"time" 0.000000e+00
"group"
0.000000e+00
0.000000e+00
0.000000e+00
0.000000e+00
"time" 1.000000e-09
```

```
C:\Users\TopGun\netlist.csdf - Notepad++
File Edit Search View Encoding Lan
netlist.sp x netlist.csdf x
1 #H
2 SOURCE='SYMSPICE'
3 TITLE='* # file name: C:
4 SUBTITLE=''
5 TIME='19:39:10' DATE='28
6 ANALYSIS='TR' TEMPERATUR
7 COMPLEXVALUES='NO' FORMA
8 XBEGIN=' 0.000000e+000'
9 NODES=' 3'
10 #N 'v(1)' 'v(2)' 'i(v1)'
11
12 #C 0.00000000e+000
13 -0.00000000e+000
14
15 #C 2.00000000e-011
16 -1.99999998e-005
17
18 #C 4.00000000e-011
19 -3.99999996e-005
20
21 #C 6.00000000e-011
22 -5.99999994e-005
23
24 #C 8.00000000e-011
25 -7.99999992e-005
26
27 #C 1.00000000e-010
Ni length: 4 942 lines: 164 Ln: 1
```

```
C:\Users\TopGun\netlist.nut - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
netlist.sp x netlist.csdf x netlist.nut x
1 Title: // Generated for: SymSpice
2 Date: 7:40:14 PM, Sat Sep 28, 2019
3 Plotname: Transient Analysis `tran1': time = (0.s -> 1.e-09s)
4 Flags: real
5 No. Variables: 4
6 No. Points: 50
7 Variables:
8 0 time s
9 1 1 voltage
10 2 2 voltage
11 3 v1:p current
12 Values:
13 0 0
14 0
15 0
16 0
17 1 2.00000000e-11
18 2.00000000e-02
19 1.99999998e-10
20 -1.99999998e-05
21 2 4.00000000e-11
22 4.00000000e-02
23 3.99999996e-10
24 -3.99999996e-05
25 3 6.00000000e-11
26 6.00000000e-02
27 5.99999994e-10
Ni length: 3 780 lines: 213 Ln: 1 Col: 1 Sel: 0|0 Unix (LF) UTF-8 INS
```

Текстовые и бинарные данные в САПР (5)

```
C:\Users\TopGun\netlist.nut - Notepad++  
File Edit Search View Encoding Language Settings  
netlist.sp netlist.csdif netlist.nut  
1 Title: // Generated for: SymSpice  
2 Date: 7:40:14 PM, Sat Sep 28, 2019  
3 Plotname: Transient Analysis `tranl  
4 Flags: real  
5 No. Variables: 4  
6 No. Points: 50  
7 Variables:  
8 0 time s  
9 1 1 voltage  
10 2 2 voltage  
11 3 vl:p current  
12 Values:  
13 0 0  
14 0  
15 0  
16 0  
17 1 2.00000000e-11  
18 2.00000000e-02  
19 1.99999998e-10  
20 -1.99999998e-05  
21 2 4.00000000e-11  
22 4.00000000e-02  
23 3.99999996e-10  
24 -3.99999996e-05  
25 3 6.00000000e-11  
26 6.00000000e-02  
27 5.99999994e-10  
Ni length : 3 780 lines : 213 Ln: 1 Col: 1 Sel: 0
```

```
E:\SymicaFree\test_NOT_nutbin.tr0  
Title: // Generated for: SymSpice  
Date: 1:22:05 PM, Thu Oct 21, 2021  
Plotname: DC Analysis `dc1': vin:dc = (0.V -> 5.0 V)  
Flags: real  
No. Variables: 6  
No. Points: 501  
Variables:  
0 volt V  
1 in voltage  
2 out voltage  
3 vcc voltage  
4 vcc:p current  
5 vin:p current  
Binary:  
@!!яая2w@! S!↓ГгГwюБ ?,,z6G@!{?,,z6G@!{@!!яая2dS@! S!↓Г!§L8Б ?"z6G@!{?"z6G@!{@!!яая1h†@!  
=pJЧ  
?.  
=pJЧ  
@!!яая.Б&@! S!↓Гъb:Б ?N°TMTMTMTMTMль?N°TMTMTMTMTMль@!!яая.'€@! S!↓Г-→.йБ ?j(xBU\)?j(xBU\}@!!яая-¶¶@! S!↓Г -y'Б  
=pH?AJЧ  
=pH@!!яая,У_@! S!↓Г""x1#Б ?Бл...▲ёQм?Бл...▲ёQм@!!яэ-)м±@! s?L3п=rPБ ?Г333333?Г333333@!!япLс °@! s1(+@_+  
=pJЧ  
?З  
=pJЧ  
@!!яя}TË+@! sKМmbOXБ ?IQл...▲ёR?IQл...▲ёR@!!я--At@! sГъwчРyБ ?Й°TMTMTMTMTMль?Й°TMTMTMTMTMль@!!юbl!!Ik@! s°2н:W%hБ  
=q?HрJЧ  
=q@!!эЦ §гГ@! sщ.рmдКБ ?OёQл...▲ё?OёQл...▲ё@!!э♦y=h+@! sГ~,гаA0Б ?P ?P @!!ьl♥knÿ@! s3Yiьц)g  
=pH?PJЧ  
=pH@!!ыдëwш8@! sЛKLEфдБ ?CG@!z6H?CG@!z6H@!!ыдшÏ@8@! sP  
§Kгг'Б ?Сл...▲ёQм?Сл...▲ёQм@!!ьJE&φ@! sТБ°Г"~Б ?ТЦ\ (xBU)?ТЦ\ (xBU@!!щwIЦь!!@! sX▼льfPMБ ?У333333?У333  
=pJЧ?УЧ  
=pJЧ@!!чГ]↑kx@! sБгjф▼-ьБ ?Фz6G@!{?Фz6G@!{@!!цГИ!P»@! sЮ¶ÿcj†Б ?X▲ёQл...▼?X▲ёQл...▼@!!x-▲♥ -@! sa-0X]  
=pJЧ
```

Текстовые и бинарные данные в САПР (6)

```
E:\SymicaFree\test_NOT_nutascii.tr0
Title: // Generated for: SymSpice
Date: 1:22:06 PM, Thu Oct 21, 2021
Plotname: DC Analysis `dc1': vin:dc = (0.V -> 5.0 V)
Flags: real
No. Variables: 6
No. Points: 501
Variables:
      0      volt      V
      1      in       voltage
      2      out      voltage
      3      vcc      voltage
      4      vcc:p    current
      5      vin:p    current

Values:
0
0
4.999999999e+00
5
-1.005000000e-11
0
1
1.000000000e-02
1.000000000e-02
4.999999999e+00
5
-1.005000001e-11
0
```

40 600 Байт

```
E:\SymicaFree\test_NOT_nutbin.tr0
Title: // Generated for: SymSpice
Date: 1:22:05 PM, Thu Oct 21, 2021
Plotname: DC Analysis `dc1': vin:dc = (0.V -> 5.0 V)
Flags: real
No. Variables: 6
No. Points: 501
Variables:
      0      volt      V
      1      in       voltage
      2      out      voltage
      3      vcc      voltage
      4      vcc:p    current
      5      vin:p    current

Binary:
      0      volt      V
      1      in       voltage
      2      out      voltage
      3      vcc      voltage
      4      vcc:p    current
      5      vin:p    current

@!!яая2w@! S!↓Ггггюб ?,,z6G@!{?,z6G@!{!я
=pJЧ
?-
=pJЧ
@!!яая.Б&@! S!↓Ггггб:Б ?N@!!яая.'@! S!↓
=pJЧ?AJЧ
=pJЧ@!!яая,У_@! S!↓Ггггх1#Б ?Бл...▲ёQм?Бл...▲ёQм@!!яэ-)м±@!
=pJЧ
?3
=pJЧ
@!!яг}TЭ+@! sKМмбOXБ ?IQл...▲ёR?IQл...▲ёR@!!я--At@! sГ
=q?HрJЧ
```

24 333 Байт

Текстовые и бинарные данные в САПР (8)

```
view simple1_msh.grd - Far 3.0.3000 x86
E:\Samples_tplot\3D_first\Simple\simple1_msh.grd
DF-ISE text

# filename: simple1_msh.grd
# written by the library Delaunay
Info {
  version      = 1.1
  type         = grid
  dimension    = 3
  nb_vertices  = 45
  nb_edges     = 180
  nb_faces     = 232
  nb_elements  = 112
  nb_regions   = 4
  regions      = [ "bott" "top" "Nplus" "pSubs" ]
  materials    = [ Contact Contact Silicon Silicon ]
}
Data {
  CoordSystem {
    translate = [ 0 0 0 ]
    transform = [ 1 0 0 0 1 0 0 0 1 ]
  }
  Vertices ( 45) {
    0 0 0
    2.5 0 0
    2.5 2.5 0
    0 2.5 0
    0 0 2.5
```

```
view simple1_msh.dat - Far 3.0.3000 x86
E:\Samples_tplot\3D_first\Simple\simple1_msh.dat
DF-ISE text

# filename: simple1_msh.dat
# written by the library Delaunay
Info {
  version      = 1
  type         = dataset
  dimension    = 3
  nb_vertices  = 45
  nb_edges     = 180
  nb_faces     = 232
  nb_elements  = 112
  nb_regions   = 4
  datasets     = [ "DopingConcentration" "BoronActiveConce"
  functions    = [ DopingConcentration BoronActiveConcent
}
Data {
  Dataset ("DopingConcentration") {
    function = DopingConcentration
    type     = scalar
    dimension = 1
    location = vertex
    validity = ["Nplus"]
    Values ( 27) {
      1e+20
      1e+20
      1e+20
```

Как Linux определяет тип файла?

```
mc [topgun@localhost.localdomain]:/usr/share/misc
/usr/share/misc/magic 280409/904K 30%
0 string \177ELF ELF
!;strength *2
>4 byte 0 invalid class
>4 byte 1 32-bit
>4 byte 2 64-bit
>5 byte 0 invalid byte order
>5 byte 1 LSB
>>0 use elf-le
>5 byte 2 MSB
>>0 use \^elf-le
>7 byte 0 (SYSV)
>7 byte 1 (HP-UX)
>7 byte 2 (NetBSD)
>7 byte 3 (GNU/Linux)
>7 byte 4 (GNU/Hurd)
>7 byte 5 (86Open)
>7 byte 6 (Solaris)
>7 byte 7 (Monterey)
>7 byte 8 (IRIX)
>7 byte 9 (FreeBSD)
>7 byte 10 (Tru64)
1Help 2Wrap 3Quit 4Hex 5Goto 6 7Search 8Raw
```

```
mc [topgun@localhost.localdomain]:/usr/bin
/usr/bin/gcc-7 212/212 100%
/usr/bin/gcc-7: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=60bf3bf11cdfaae7fef244d3435f92193781f82e, stripped
1Help 2UnWrap 3Quit 4Hex 5Goto 6
```

```
mc [topgun@localhost.localdomain]:/usr/bin
/usr/bin/gcc-7 0x00000007 0%
00000000 7F 45 4C 46 | 01 01 01 30 | 00 00 00 00 | 00 00 00 00 | .ELF.....
00000010 02 00 03 00 | 01 00 00 00 | CD 84 0A 08 | 34 00 00 00 | .....4...
00000020 70 3A 0F 00 | 00 00 00 00 | 34 00 20 00 | 0A 00 28 00 | p:.....4...
00000030 20 00 1F 00 | 06 00 00 00 | 34 00 00 00 | 34 80 04 08 | .....4...
00000040 34 80 04 08 | 40 01 00 00 | 40 01 00 00 | 05 00 00 00 | 4...@...@...
00000050 04 00 00 00 | 03 00 00 00 | 74 01 00 00 | 74 81 04 08 | .....t...t...
00000060 74 81 04 08 | 13 00 00 00 | 13 00 00 00 | 04 00 00 00 | t.....
00000070 01 00 00 00 | 01 00 00 00 | 00 00 00 00 | 00 80 04 08 | .....
00000080 00 80 04 08 | E0 18 0F 00 | E0 18 0F 00 | 05 00 00 00 | .....
00000090 00 10 00 00 | 01 00 00 00 | 84 25 0F 00 | 84 B5 13 08 | .....%.....
000000A0 84 B5 13 08 | BC 13 00 00 | 6C 37 00 00 | 06 00 00 00 | .....17.....
000000B0 00 10 00 00 | 02 00 00 00 | E0 2E 0F 00 | E0 BE 13 08 | .....
000000C0 E0 BE 13 08 | F8 00 00 00 | F8 00 00 00 | 06 00 00 00 | .....
000000D0 04 00 00 00 | 04 00 00 00 | 88 01 00 00 | 88 81 04 08 | .....
000000E0 88 81 04 08 | 44 00 00 00 | 44 00 00 00 | 04 00 00 00 | ...D...D...
000000F0 04 00 00 00 | 07 00 00 00 | 84 25 0F 00 | 84 B5 13 08 | .....%.....
00000100 84 B5 13 08 | 00 00 00 00 | 08 00 00 00 | 04 00 00 00 | .....
00000110 04 00 00 00 | 50 E5 74 64 | 80 B5 0C 00 | 80 35 11 08 | ...P..td...5..
00000120 80 35 11 08 | 64 36 00 00 | 64 36 00 00 | 04 00 00 00 | .5..d6..d6.....
00000130 04 00 00 00 | 51 E5 74 64 | 00 00 00 00 | 00 00 00 00 | ...Q..td.....
00000140 00 00 00 00 | 00 00 00 00 | 00 00 00 00 | 06 00 00 00 | .....
00000150 10 00 00 00 | 52 E5 74 64 | 84 25 0F 00 | 84 B5 13 08 | ...R..td.%.....
1Help 2Edit 3Quit 4Ascii 5Goto 6Save 7HxSrch 8Parse 9Format10Quit
```

Распараллеливание вычислений: процессы и потоки

Поток – (thread, нить) – базовая единица загрузки процессора



ПРОЦЕСС



ПОТОК 1

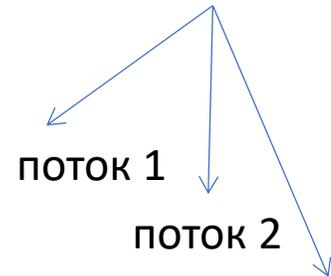


Сегмент кода
Сегмент данных

Регистры
Стек
+
идентификатор



ПРОЦЕСС



ПОТОК 1

ПОТОК 2

ПОТОК 3

Распараллеливание расчётов в современном C++

```
#include <thread>

void print_0() {
    for (int i = 0; i < 1000; ++i)
        printf("0");
}

void print_1() {
    for (int i = 0; i < 1000; ++i)
        printf("1");
}

int main() {
    std::thread thread1(print_0);
    std::thread thread2(print_1);

    thread1.join();
    thread2.join();
    return 0;
}
```

```
#include <pthread.h>

void *print_0(void *args) {
    for (int i = 0; i < 1000; ++i)
        printf("0");
    return NULL;
}

void *print_1(void *args) {
    for (int i = 0; i < 1000; ++i)
        printf("1");
    return NULL;
}

int main() {
    pthread_t thread1, thread2;

    pthread_create(&thread1, NULL, print_0, NULL);
    pthread_create(&thread2, NULL, print_1, NULL);

    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);

    return 0;
}
```

Распараллеливание расчётов: взаимодействие потоков

```
std::vector<int> values;

void print_0() {
    for (int i = 0; i < 10000; i++)
        printf("0");
    values.push_back(0);
}

void print_1() {
    for (int i = 0; i < 10000; i++)
        printf("1");
    values.push_back(1);
}

int main() {
    thread thread1(print_0);
    thread thread2(print_1);

    thread1.join();
    thread2.join();
    return 0;
}
```



```
std::vector<int> values;
std::mutex      m;

void print_0() {
    for (int i = 0; i < 10000; i++)
        printf("0");
    m.lock();
    values.push_back(0);
    m.unlock();
}

void print_1() {
    for (int i = 0; i < 10000; i++)
        printf("1");
    m.lock();
    values.push_back(1);
    m.unlock();
}
```

Межпроцессное взаимодействие



Файлы

Сообщения и сигналы

Сокеты

Разделяемая память

IPC: файлы (FS change notification)

```
HANDLE FindFirstChangeNotificationA(  
    LPCSTR lpPathName,  
    BOOL bWatchSubtree,  
    DWORD dwNotifyFilter  
);
```

Заголовочный файл: fileapi.h

[in] dwNotifyFilter

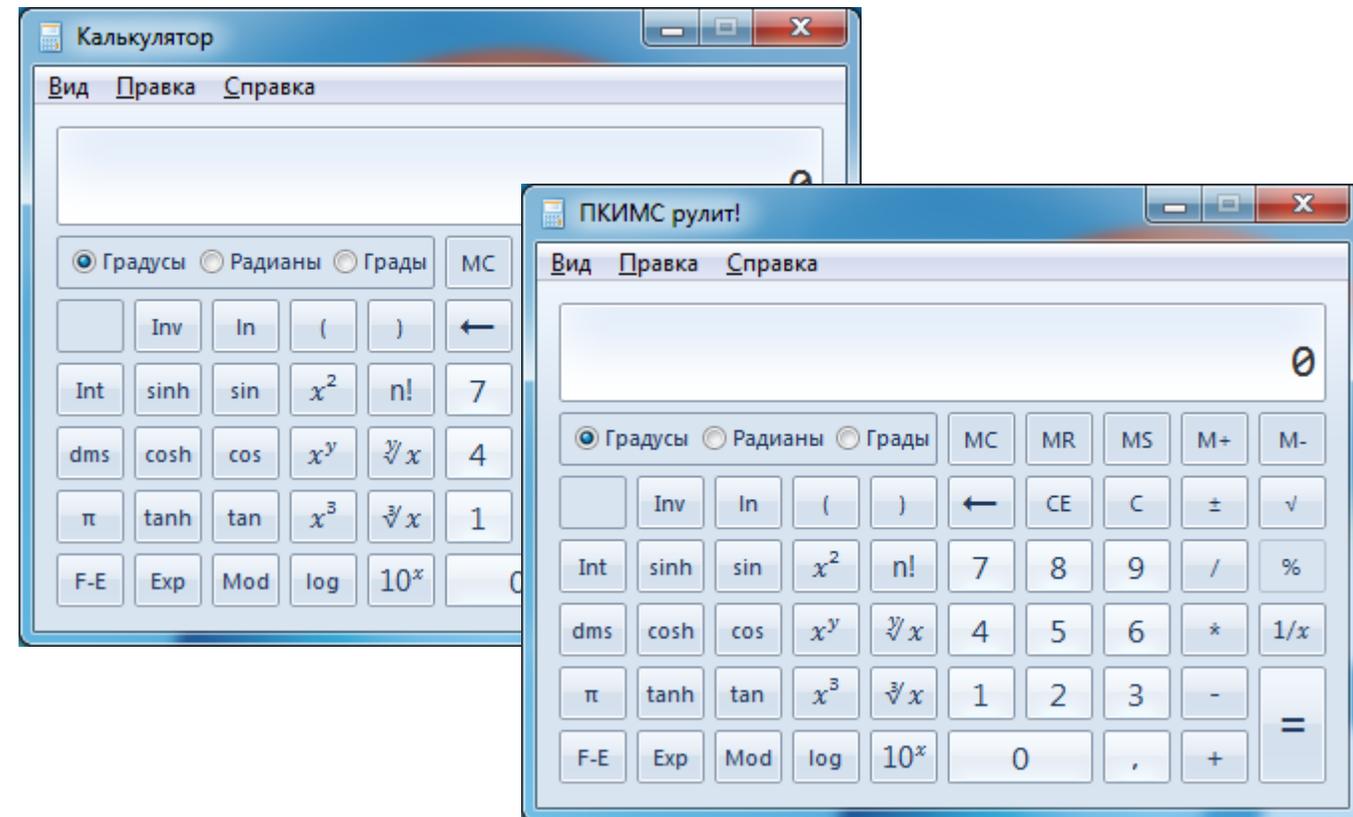
The filter conditions that satisfy a change notification wait. This parameter can be one or more of the following values.

Value	Meaning
FILE_NOTIFY_CHANGE_FILE_NAME 0x00000001	Any file name change in the watched directory or subtree causes a change notification wait operation to return. Changes include renaming, creating, or deleting a file name.
FILE_NOTIFY_CHANGE_DIR_NAME 0x00000002	Any directory-name change in the watched directory or subtree causes a change notification wait operation to return. Changes include creating or deleting a directory.
FILE_NOTIFY_CHANGE_ATTRIBUTES 0x00000004	Any attribute change in the watched directory or subtree causes a change notification wait operation to return.
FILE_NOTIFY_CHANGE_SIZE 0x00000008	Any file-size change in the watched directory or subtree causes a change notification wait operation to return. The operating system detects a change in file size only when the file is written to the disk. For operating systems that use extensive caching, detection occurs only when the cache is sufficiently flushed.
FILE_NOTIFY_CHANGE_LAST_WRITE 0x00000010	Any change to the last write-time of files in the watched directory or subtree causes a change notification wait operation to return. The operating system detects a change to the last write-time only when the file is written to the disk. For operating systems that use extensive caching, detection occurs only when the cache is sufficiently flushed.
FILE_NOTIFY_CHANGE_SECURITY 0x00000100	Any security-descriptor change in the watched directory or subtree causes a change notification wait operation to return.

IPC: сообщения (signals, messages)



```
wchar_t str[] = L"ПКИМС рулит!";  
HWND hWnd = FindWindow(nullptr, L"Калькулятор");  
  
if (nullptr != hWnd)  
    SendMessage(hWnd, WM_SETTEXT, 0, (LPARAM)str);
```

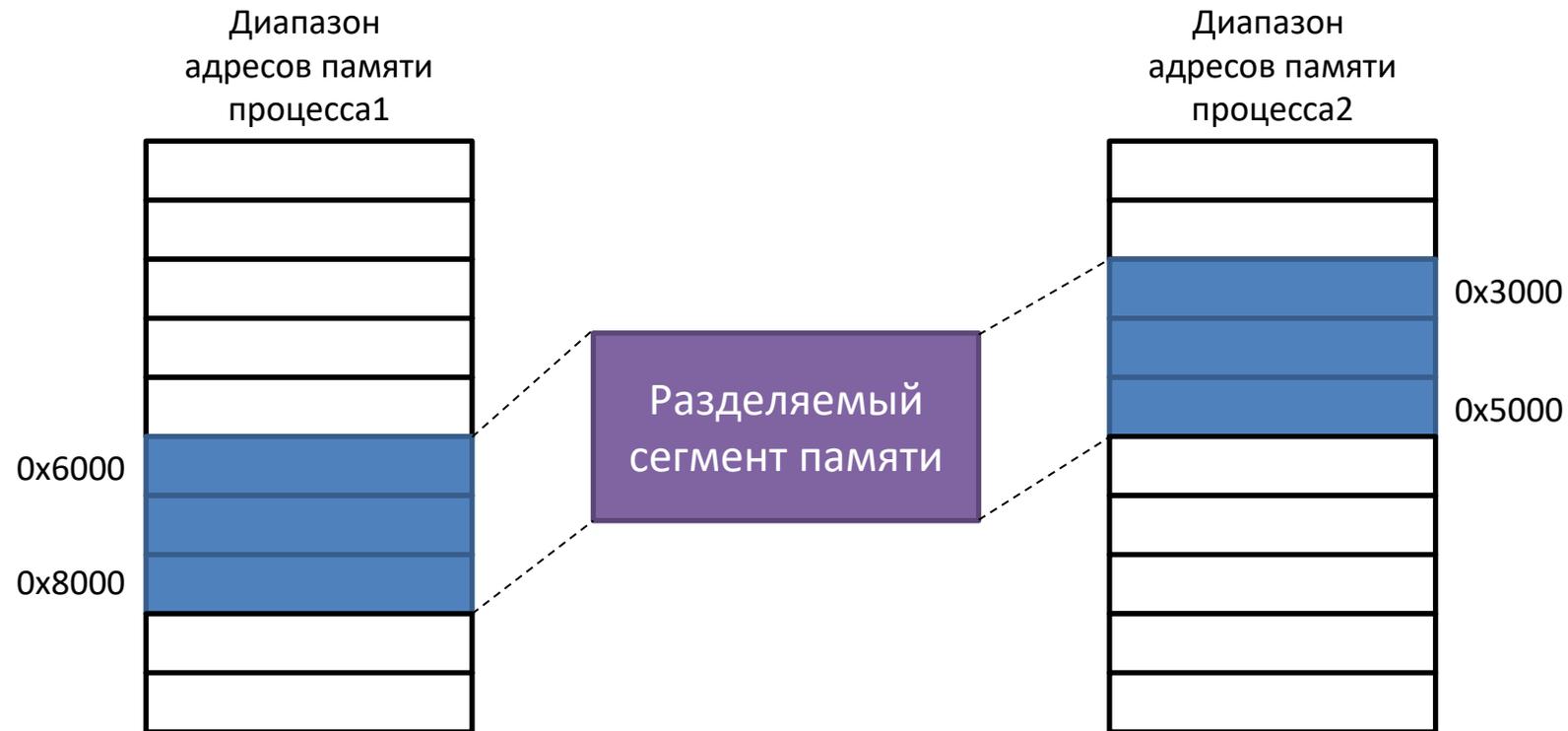


IPC: сокеты (sockets)

```
1 import os
2 import socket
3
4 SOCKET_FILE = './echo.socket'
5
6 if os.path.exists(SOCKET_FILE):
7     os.remove(SOCKET_FILE)
8
9 print("Открываем UNIX сокет...")
10 server = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)
11 server.bind(SOCKET_FILE)
12
13 print("Слушаем...")
14 while True:
15     datagram = server.recv(1024)
16     if not datagram:
17         break
18     else:
19         print("-" * 20)
20         print(datagram)
21         if b"DONE" == datagram:
22             break
23 print("-" * 20)
24 print("Выключение...")
25 server.close()
26 os.remove(SOCKET_FILE)
27 print("Выполнено")
```

```
1 import os
2 import socket
3
4 SOCKET_FILE = './echo.socket'
5
6 print("Подключение...")
7 if os.path.exists(SOCKET_FILE):
8     client = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)
9     client.connect(SOCKET_FILE)
10    print("Выполнено.")
11    print("Ctrl-C чтобы выйти.")
12    print("Отправьте 'DONE' чтобы выключить сервер.")
13    while True:
14        try:
15            x = input("> ") # for py2 use raw_input
16            if "" != x:
17                print("ОТПРАВЛЕНО: %s" % x)
18                client.send(x.encode('utf-8'))
19                if "DONE" == x:
20                    print("Выключение.")
21                    break
22        except KeyboardInterrupt as k:
23            print("Выключение.")
24            break
25    client.close()
26 else:
27    print("Не могу соединиться!")
28 print("Выполнено")
```

IPC: разделяемая память (shared memory) (1)



IPC: разделяемая память (shared memory) (2)

```
#include <iostream>
#include <sys/ipc.h>
#include <sys/shm.h>

using namespace std;

int main() {
    key_t key = ftok("shmfile", 65);

    int shmid = shmget(key, 1024, 0666 | IPC_CREAT);

    char *str = (char*) shmat(shmid, (void *)0, 0);

    cout << "Write Data : ";
    gets(str);

    cout << "Data written in memory: " << str << endl;

    shmdt(str);

    return 0;
}
```

```
#include <iostream>
#include <sys/ipc.h>
#include <sys/shm.h>

using namespace std;

int main() {
    key_t key = ftok("shmfile", 65);

    int shmid = shmget(key, 1024, 0666 | IPC_CREAT);

    char *str = (char*) shmat(shmid, (void *)0, 0);

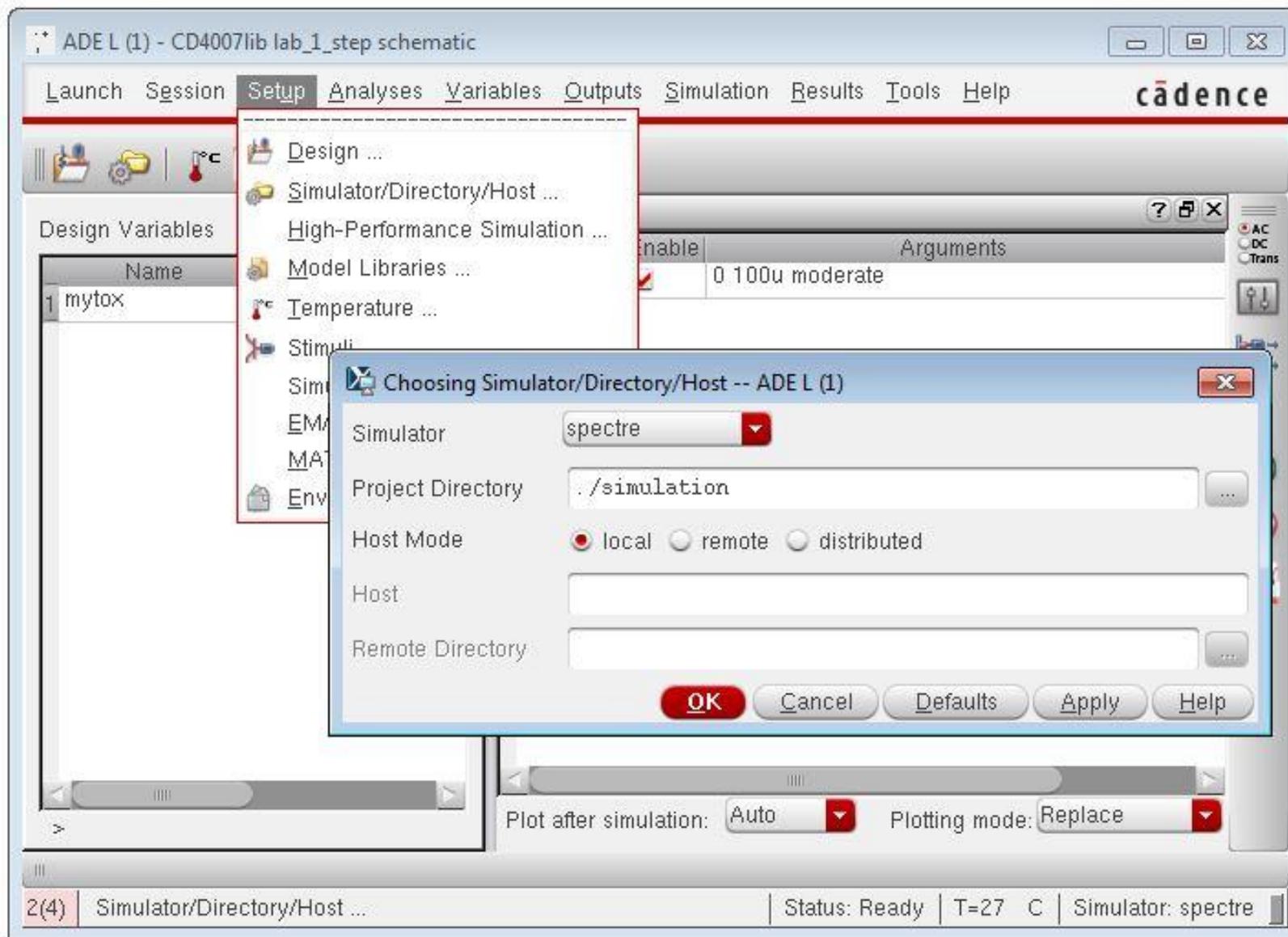
    cout << "Data read from memory: " << str << endl;

    shmdt(str);

    shmctl(shmid, IPC_RMID, NULL);

    return 0;
}
```

Нелокальное выполнение расчётов (1)



Нелокальное выполнение расчётов (2)





Онлайн симуляторы: цифровой этап проектирования (1)

The screenshot displays the edaplayground.com web interface. The browser address bar shows the URL `edaplayground.com`. The page header includes the EDA playground logo, navigation buttons for Run, Save, and Playgrounds, and a login link. A notification banner indicates routine maintenance from 2019-10-09 04:15:00 (UTC) to 12:15:00 (UTC). The main workspace is divided into two code editors: `testbench.sv` and `design.sv`. The `testbench.sv` editor contains the following code:

```
1 // Code your testbench here
2 // or browse Examples
3
```

The `design.sv` editor contains the following code:

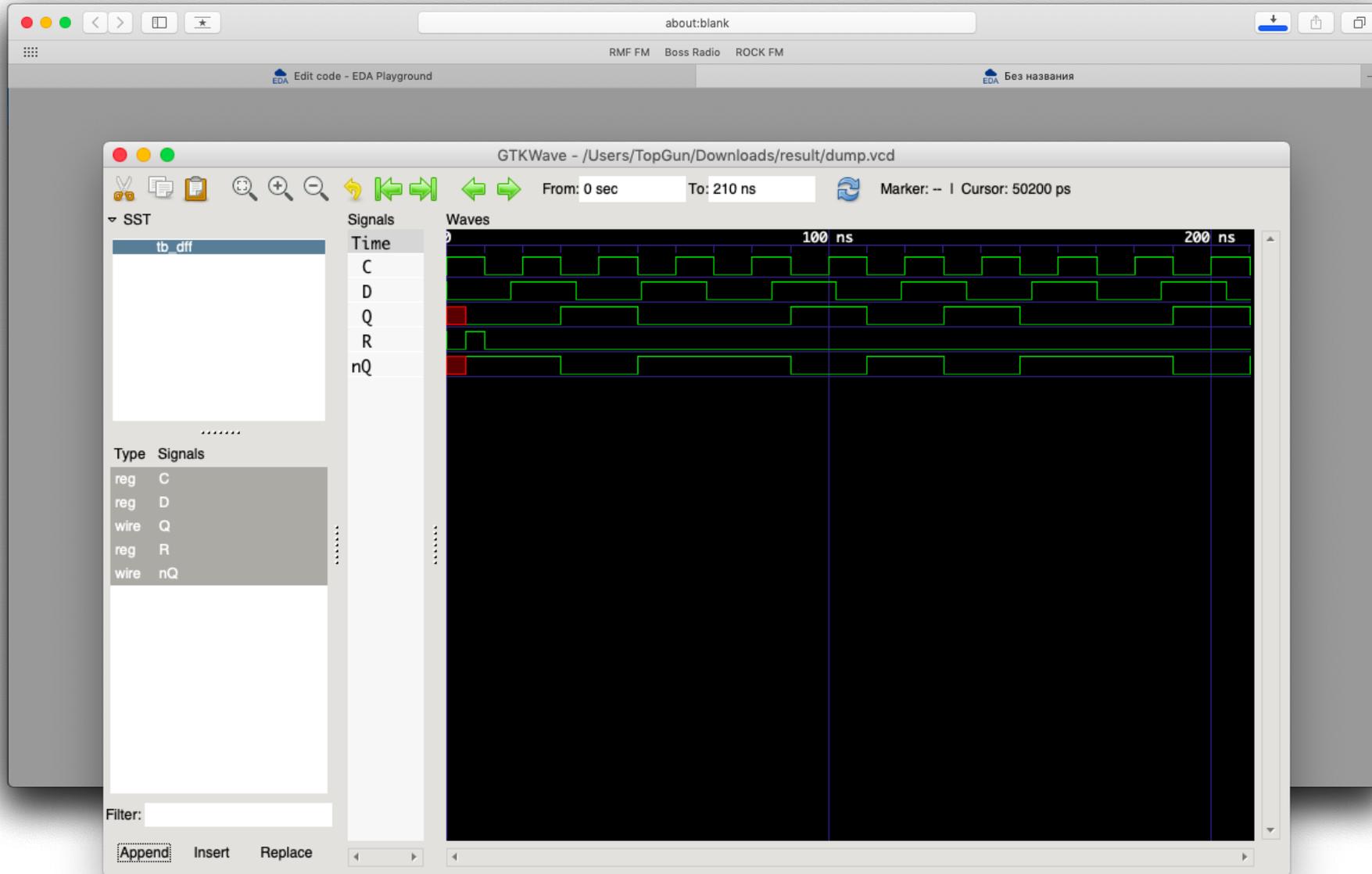
```
1 // Code your design here
2
```

Below the code editors, there are buttons for Log and Share, a text input field for adding a title, a dropdown menu for visibility (set to Public), and a Save button. A rich text editor is visible below, with a toolbar containing Bold (B), Italic (I), Highlight (H), Quote, List, Link, and other icons. A placeholder text reads: "A short description will be helpful for you to remember your playground's details".

The left sidebar contains navigation and configuration options:

- Languages & Libraries**
 - Testbench + Design**
 - SystemVerilog/Verilog
 - UVM / OVM**
 - None
 - Other Libraries**
 - None
 - OV_L 2.8.1
 - SVUnit 2.11
 - SVAUnit 3.0
 - Enable TL-Verilog
 - Enable Easier UVM
 - Enable VUnit
- Tools & Simulators**
 - Select...
 - Open EPWave after run
 - Download files after run
- Examples**
- Community**
 - Collaborate
 - Forum
 - Follow @edaplayground

Онлайн симуляторы: цифровой этап проектирования (2)



Онлайн симуляторы: схемотехническое моделирование (1)

The screenshot displays the CircuitLab online circuit simulator interface. The browser address bar shows the URL www.circuitlab.com/editor/. The interface includes a top navigation bar with 'File', 'Edit', 'Run', and 'Help' menus. A user notification reads 'Hi there, Dmitry Bulakh (TopGun_DICD)' and a warning states 'Warning: unsaved changes! Now editing: Unnamed Circuit'.

On the left side, there is a component library with a search bar and several categories:

- Essentials:** Includes 'NAME NODE' and 'WIRE' tools.
- Ideal Sources:** Contains icons for DC and AC voltage sources.
- Passive Elements:** Contains icons for resistors, capacitors, and inductors.
- Signal Sources:** Contains icons for various signal generators.
- Operational Amplifiers:** Contains icons for op-amp models.
- Diodes:** Contains icons for different diode types.

The main workspace shows a circuit diagram on a grid background. The circuit consists of a square wave voltage source labeled 'V1 square 50 MHz' connected in series with a resistor labeled 'R1 1 kΩ'. This series combination is connected to a capacitor labeled 'C1 1 pF'. The other end of the capacitor is connected to a ground symbol.

At the bottom of the interface, there are buttons for '+', 'Build', and 'Simulate'. A yellow banner at the bottom center contains the text: 'Go Ad-Free! - Activate your CircuitLab membership for no more ads. Thanks! :-) circuitlab.com'. The bottom right corner shows a zoom level of '156%'.

Онлайн симуляторы: схемотехническое моделирование (2)

