МИЭТ
Национальный исследовательский университет «МИЭТ»
Кафедра ПКИМС

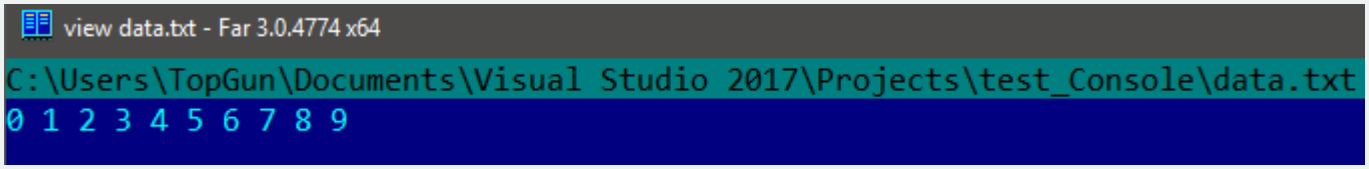
Компьютерные технологии в научных исследованиях

Семинар №3

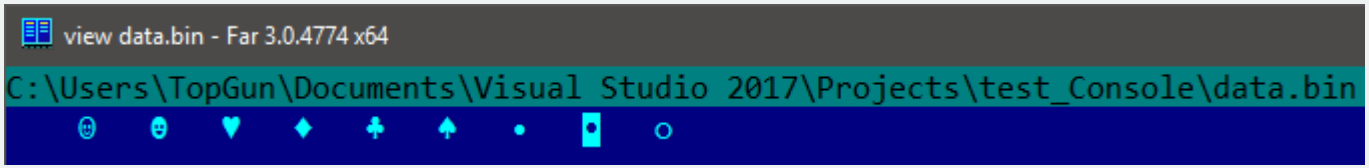
Построение вычислений

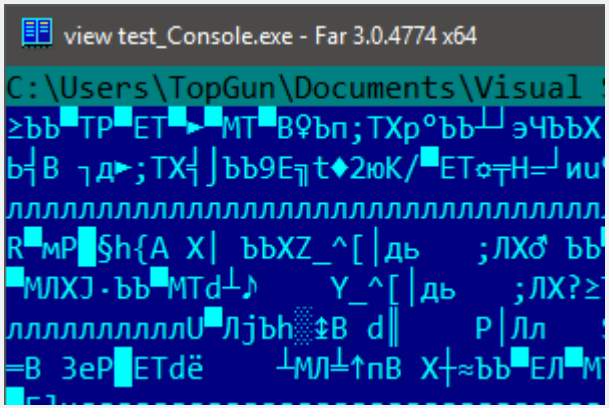
Текстовые и бинарные данные

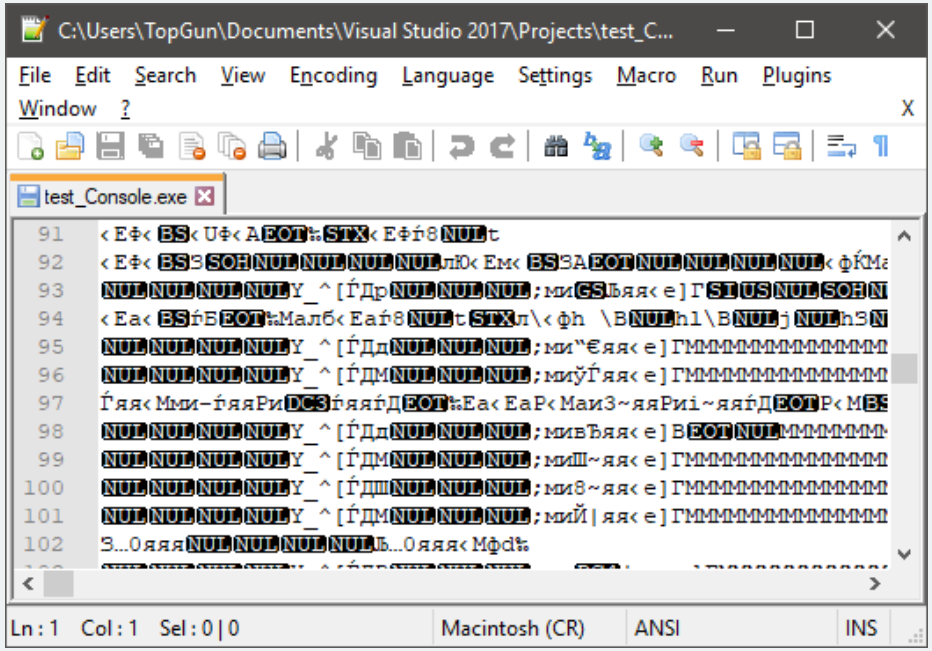
Текстовые данные (файл занимает 20 байт)



Бинарные данные (файл занимает 40 байт)

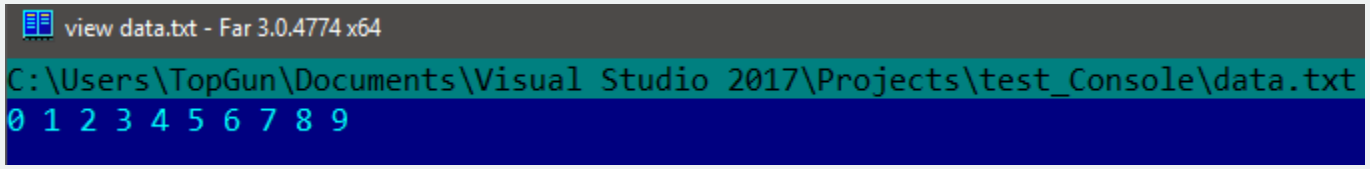






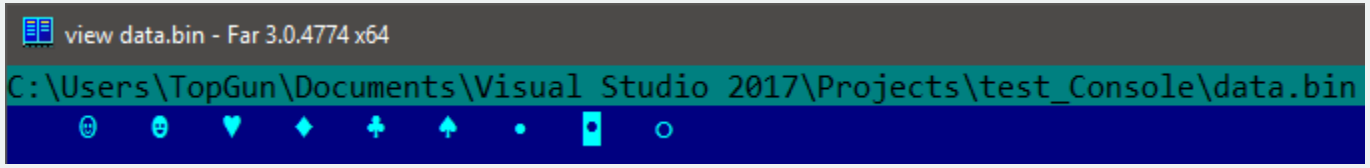
Запись текстовых данных

```
#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;
}
```



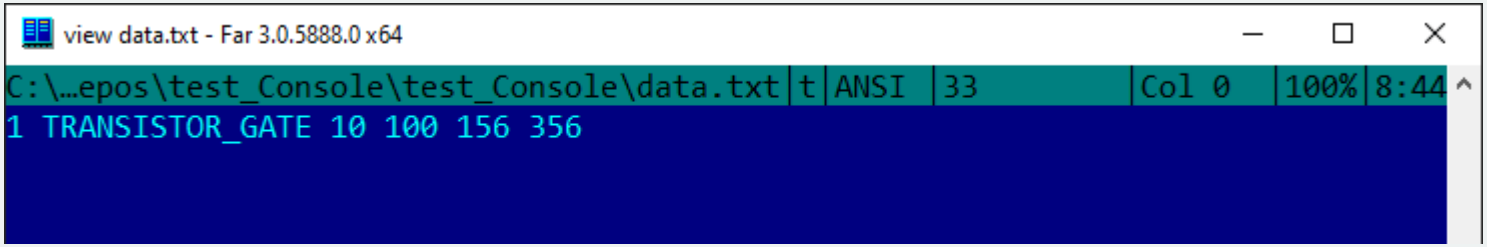
Запись бинарных данных

```
#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;
}
```



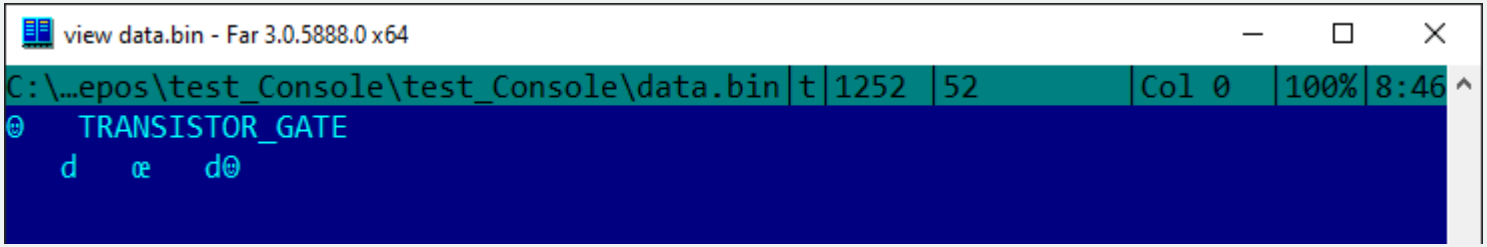
Преимущество использования бинарных данных (1)

```
struct LayoutObject {
  int  type;
  char name[16];
  int  x1, y1, x2, y2;
};

LayoutObject obj = {1, "TRANSISTOR_GATE", 10, 100, 156, 356};


fstream file_txt("data.txt", ios::out);
file_txt << obj.type << " ";
file_txt << obj.name << " ";
file_txt << obj.x1 << " " << obj.y1 << " "
         << obj.x2 << " " << obj.y2 << " ";
file_txt.close();
```



Преимущество использования бинарных данных (2)

```
struct LayoutObject {
  int  type;
  char name[16];
  int  x1, y1, x2, y2;
};

LayoutObject obj = {1, "TRANSISTOR_GATE", 10, 100, 156, 356};


fstream file_bin("data.bin", ios::out | ios::binary);
file_bin.write((char *)&obj, sizeof(LayoutObject));
file_bin.close();
```



Текстовые и бинарные данные в САПР (1)

view lab_3.tr0 - Far 3.0.4774 x64

```
C:\Users\TopGun\Documents\Visual Stud
HEADER
"PSFversion" "1.00"
"simulator" "KSI"
"runtype" "Transient Analysis"
TYPE
"node" FLOAT DOUBLE PROP(
"key" "node"
)
"branch" FLOAT DOUBLE PROP(
"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 Language Settings Macro Run Plugins Window ?

netlist.sp | netlist.csdf

```
#H
SOURCE='SYMSPICE'
TITLE='* # file name: C:\Users\TopGun\netlist                    '
SUBTITLE=''
TIME='19:39:10' DATE='28/9/2019'
ANALYSIS='TR' TEMPERATURE=' 2.500000E+001' SWEEPVAR='TIME'
COMPLEXVALUES='NO' FORMAT='1 VOLTSorAMPS;EFLOAT'
XBEGIN=' 0.000000e+000' XEND=' 1.000000e-009'
NODES='     3'
#N 'v(1)' 'v(2)' 'i(v1)'

#C     0.00000000e+000           3    0.00000000e+000   0.00000000e+000
  -0.00000000e+000

#C     2.00000000e-011           3    2.00000000e-002   1.99999998e-010
  -1.99999998e-005

#C     4.00000000e-011           3    4.00000000e-002   3.99999996e-010
  -3.99999996e-005

                                3    6.00000000e-002   5.99999994e-010

                                3    8.00000000e-002   7.99999992e-010

                                3    1.00000000e-001   9.99999990e-010
```

Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

C:\Users\TopGun\netlist.nut - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

netlist.sp | netlist.csdf | netlist.nut

```
Title: // Generated for: SymSpice
Date: 7:40:14 PM, Sat Sep 28, 2019
Plotname: Transient Analysis `tranl': time = (0.s -> 1.e-09s)
Flags: real
No. Variables: 4
No. Points: 50
Variables:
        0   time    s
        1   1   voltage
        2   2   voltage
        3   v1:p    current
Values:
 0      0
        0
        0
        0
 1      2.00000000e-11
        2.00000000e-02
        1.99999998e-10
        -1.99999998e-05
 2      4.00000000e-11
        4.00000000e-02
        3.99999996e-10
        -3.99999996e-05
 3      6.00000000e-11
        6.00000000e-02
        5.99999994e-10
```

length : 3 780 | lines : 213 | Ln : 1 | Col : 1 | Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

Текстовые и бинарные данные в САПР (2)

C:\Users\TopGun\netlist.nut - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

netlist.sp | netlist.csdf | netlist.nut

```
Title: // Generated for: SymSpice
Date: 7:40:14 PM, Sat Sep 28, 2019
Plotname: Transient Analysis `tran1': time = (0.s -> 1.e-09s)
Flags: real
No. Variables: 4
No. Points: 50
Variables:
        0   time    s
        1   1   voltage
        2   2   voltage
        3   vl:p    current
Values:
 0      0
        0
        0
        0
 1      2.00000000e-11
        2.00000000e-02
        1.99999998e-10
        -1.99999998e-05
 2      4.00000000e-11
        4.00000000e-02
        3.99999996e-10
        -3.99999996e-05
 3      6.00000000e-11
        6.00000000e-02
        5.99999994e-10
```

length : 3 780 lines : 213 Ln : 1 Col : 1 Sel

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:
```

Текстовые и бинарные данные в САПР (3)

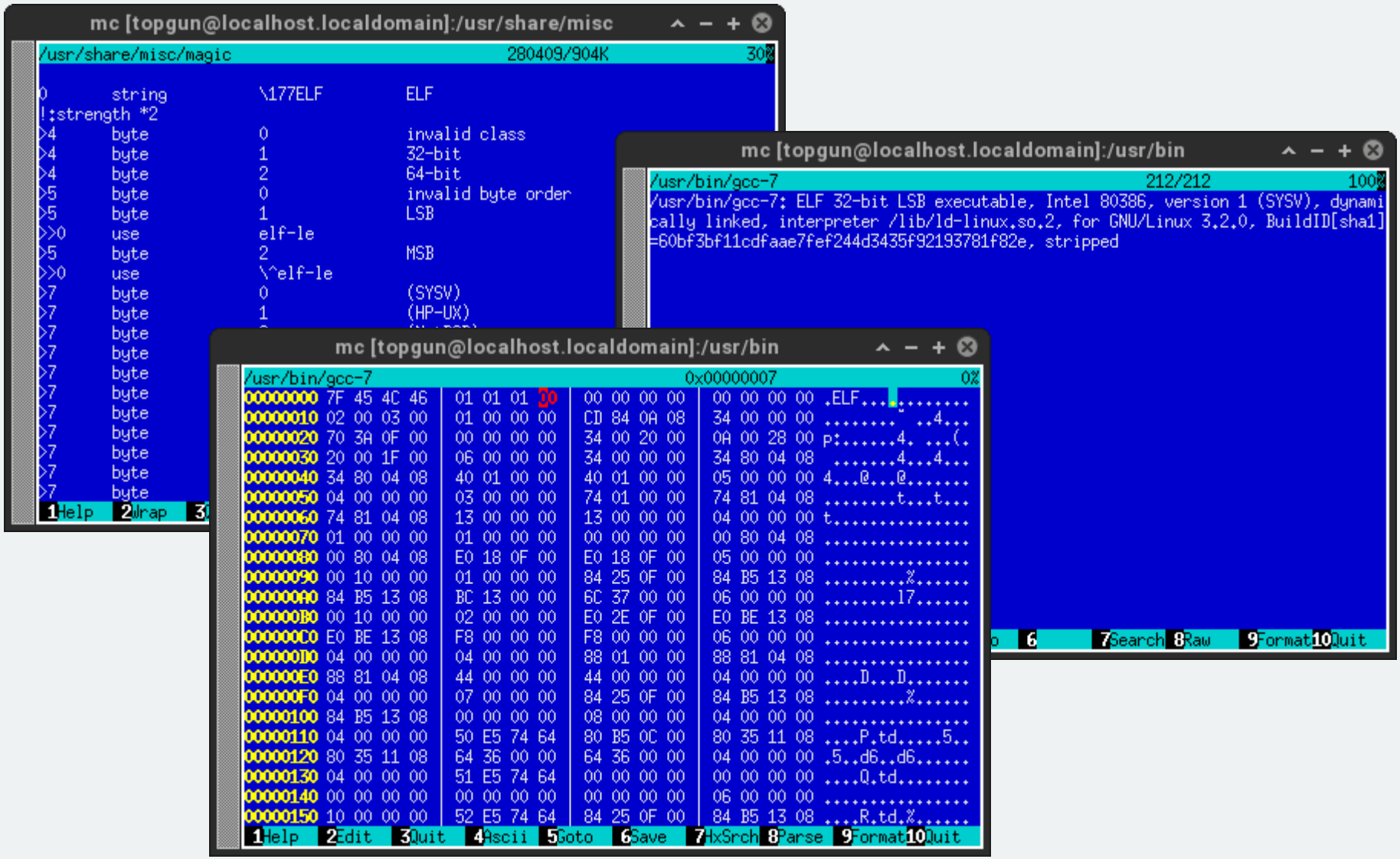
```
E:\SymicaFree\test_NOT_nutascii.tr0
Title: // Generated for: SymSpice
Date: 1:22:06 PM, Thu Oct 21, 2021
Plotname: DC Analysis `dc1': vin:dc = (0.V -> 5.0 V)
Flags: real
No. Variables: 6
No. Points: 501
Variables:
                0       volt    V
                1       in      voltage
                2       out     voltage
                3       vcc     voltage
                4       vcc:p   current
                5       vin:p   current
Values:
 0              0
                0
                4.99999999e+00
                5
                -1.00500000e-11
                0
 1              1.00000000e-02
                1.00000000e-02
                4.99999999e+00
                5
                -1.00500001e-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:
                @‼яяя2w®@¶      S¦↓Ґ¤ЃwюЂ       ?„z6G®¶{?„z6G®¶{@‼я
=pJЧ
?.
=pJЧ
@‼яяя.□&@¶      S¦↓Ґњ♀b:Ђ       ?№™™™™™™ь?№™™™™™™ь@‼яяя.'Є@¶      S¦↓
=p¤?AJЧ
=p¤@‼яяя,У_@¶      S¦↓Ґ™xl#Ђ       ?Бл…▲ёQм?Бл…▲ёQм@‼яэ¬)µ‡@¶
=pJЧ
?3
=pJЧ
@‼яr}ТЁ+@¶      sќMмЬ❸X♫Ђ       ?ИQл…▲ёR?ИQл…▲ёR@‼я--Atɑ@¶      sЃ
=q?HpJЧ
```

24 333 Байт

Как Linux определяет тип файла? (4)

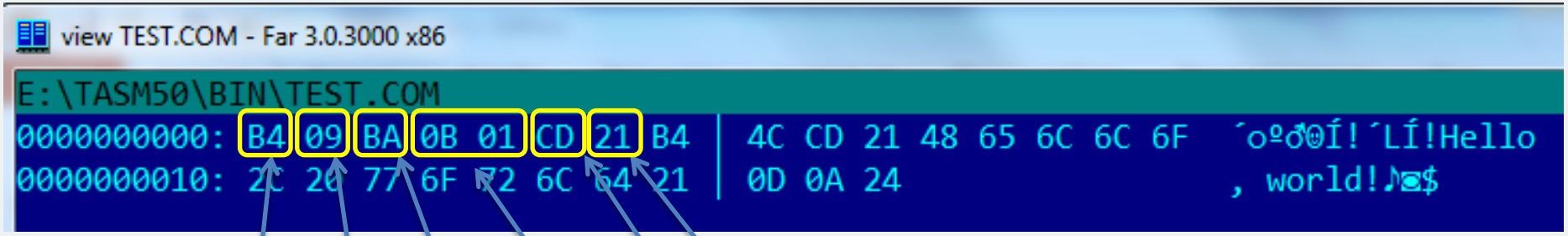


Как устроены программы? (На примере MS .COM) (1)

```c
#include <stdio.h>

int main() {
  printf("Hello, world!\n");
  return 0;
}
```

```
.model tiny
.code
org 100h
main:
        mov AH, 09h
        lea DX, string
        int 21h

        mov AH, 4Ch
        mov AL, 00h
        int 21h

        string db "Hello, world!", 0Dh, 0Ah, '$'

end main
```

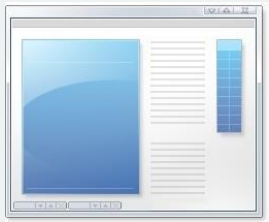
Как устроены программы? (На примере MS .COM) (3)




```
.model tiny
.code
org 100h
main:
        mov AH, 09h
        lea DX, string
        int 21h

        mov AH, 4Ch
        mov AL, 00h
        int 21h

        string db "Hello, world!", 0Dh, 0Ah, '$'

end main
```

Распараллеливание вычислений: процессы и потоки

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

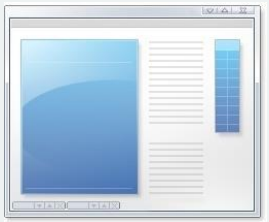


ПРОЦЕСС

поток 1

Сегмент кода
Сегмент данных

Регистры
Стек
+
идентификатор



ПРОЦЕСС

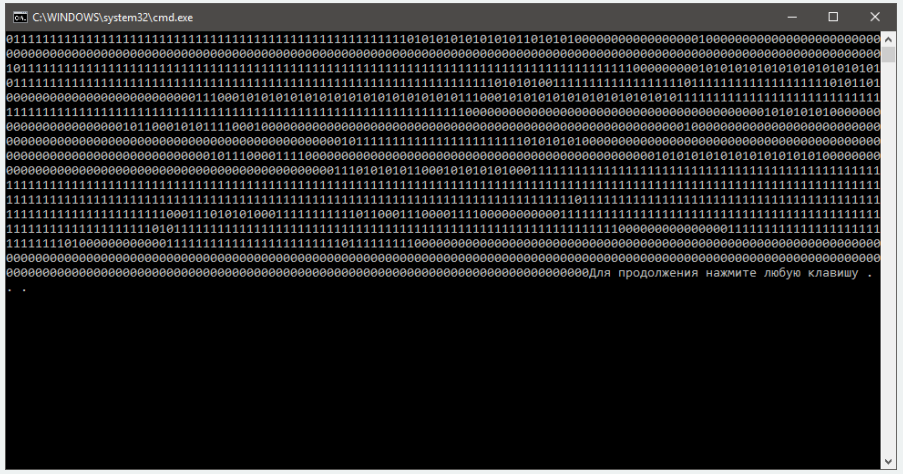
поток 1

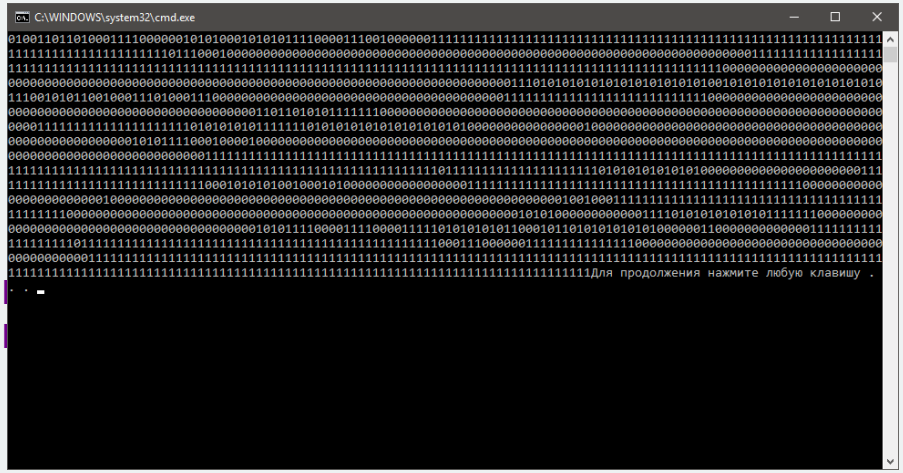
поток 2

поток 3

Распараллеливание расчётов: pthreads (2)

```c
#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: std::threads

```cpp
#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;
}
```

```cpp
#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;
}
```

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.

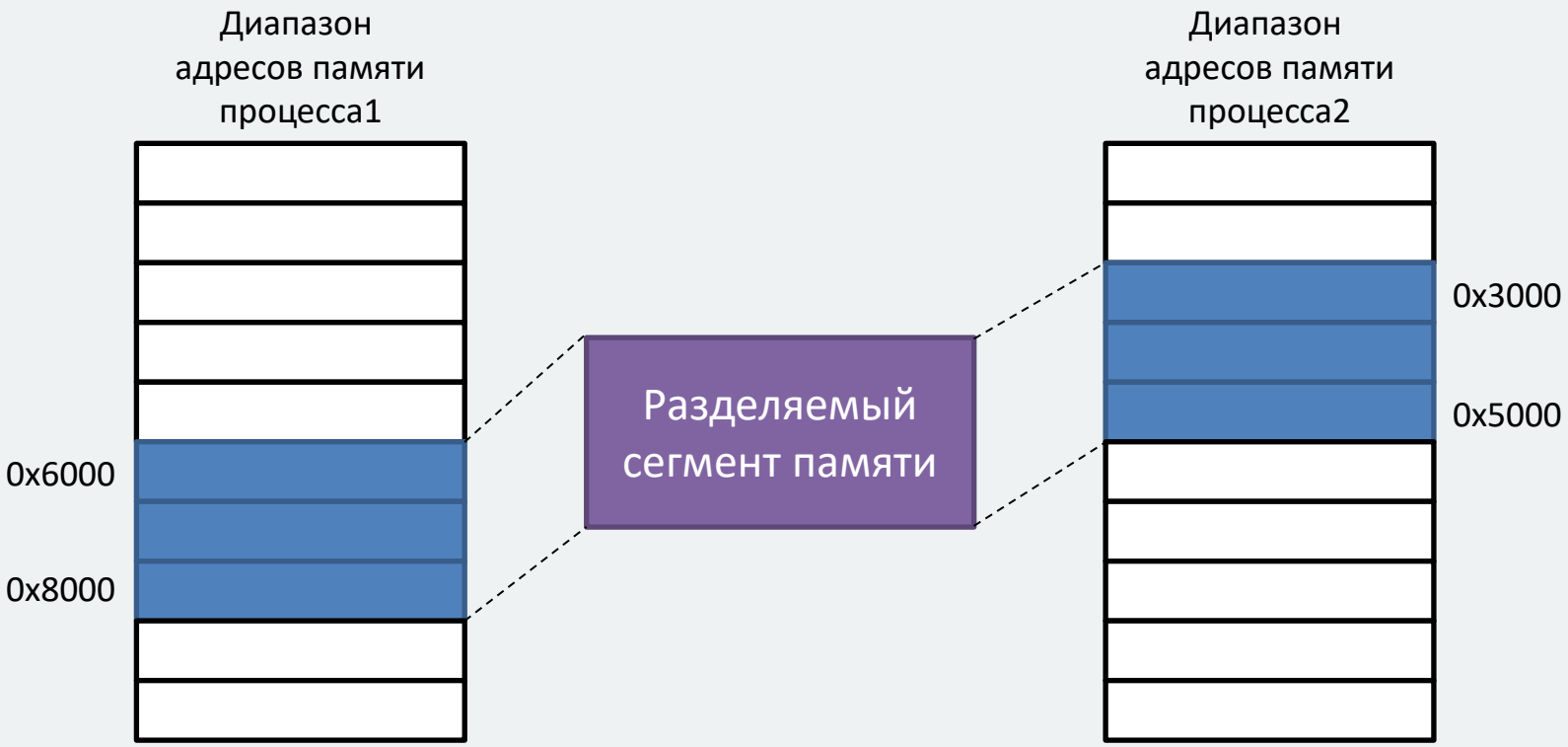
Value	Meaning
FILE_NOTIFY_CHANGE_FILE_NAME 0x00000001	Any file name change in the watched directory or subtree causes a change notification wait operation to return. Changes include renaming, creating, or deleting a file name.
FILE_NOTIFY_CHANGE_DIR_NAME 0x00000002	Any directory-name change in the watched 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) (1)


```
long __stdcall WndProcedure(HWND   hWnd,   UINT   Msg,
                            WPARAM wParam, LPARAM lParam) {
  switch (Msg) {
    case WM_CREATE:
      ...
    case WM_SIZE:
      ...
    case WM_PAINT:
      hDC = BeginPaint(hWnd, &ps);
      OnPaint(hDC);
      EndPaint(hWnd, &ps);
      break;
    case WM_LBUTTONDOWN:
      x = LOWORD(lParam);
      y = HIWORD(lParam);
      OnLButtonDown(x, y);
      break;
    ...
  }
  return 0;
}
```

```
#define WM_CREATE                0x0001
#define WM_DESTROY               0x0002
#define WM_MOVE                  0x0003
#define WM_SIZE                  0x0005
```

IPC: разделяемая память (shared memory) (1)



IPC: разделяемая память (shared memory) (3)

```cpp
#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;
}
```

```cpp
#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 read from memory: " << str << endl;

  shmdt(str);

  return 0;
}
```

Распараллеливание вычислений на GPU: CUDA (1)


```
int main() {
  int N = 100000;
  float *h_x = (float*)malloc(N * sizeof(float));
  float *h_y = (float*)malloc(N * sizeof(float));
  // Тут идёт инициализация массивов h_x и h_y

  float *d_x, *d_y;
  cudaMalloc(&d_x, N * sizeof(float));
  cudaMalloc(&d_y, N * sizeof(float));

  cudaMemcpy(d_x, h_x, N * sizeof(float), cudaMemcpyHostToDevice);
  cudaMemcpy(d_y, h_y, N * sizeof(float), cudaMemcpyHostToDevice);

  saxpy<<<(N+255)/256, 256>>>(N, 2.0f, d_x, d_y);

  cudaMemcpy(h_y, d_y, N * sizeof(float), cudaMemcpyDeviceToHost);

  cudaFree(d_x);
  cudaFree(d_y);
  free(h_x);
  free(h_y);
  return 0;
}

__global__
void saxpy(int n, float a, float *x, float *y) {
  int i = blockIdx.x*blockDim.x + threadIdx.x;
  if (i < n)
    y[i] = a * x[i] + y[i];
}
```

Распределённые вычисления

Компиляция:
$mpicc -o mpi_test mpi_test.c

Запуск:

$mpirun -np 2 ./mpi_test
$mpirun -np 5 -hosts 192.168.254.4,localhost ./mpi_test

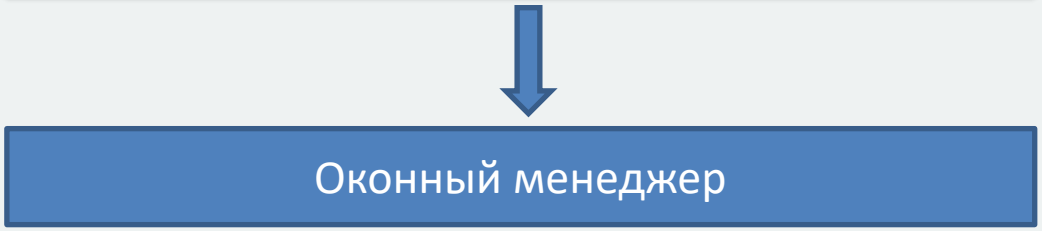
```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {
  MPI_Init(NULL, NULL);

  int world_size;
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);

  int world_rank;
  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);


  // Работаем...


  MPI_Finalize();
}
```

Нелокальное исполнение программ (1)

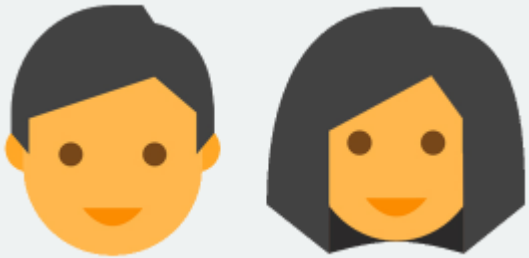
Объекты графического интерфейса системы

Элементы управления

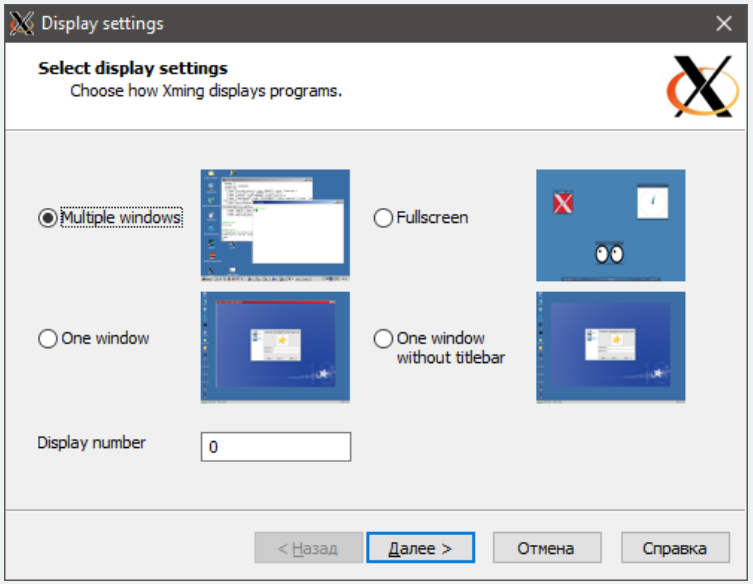
X Server



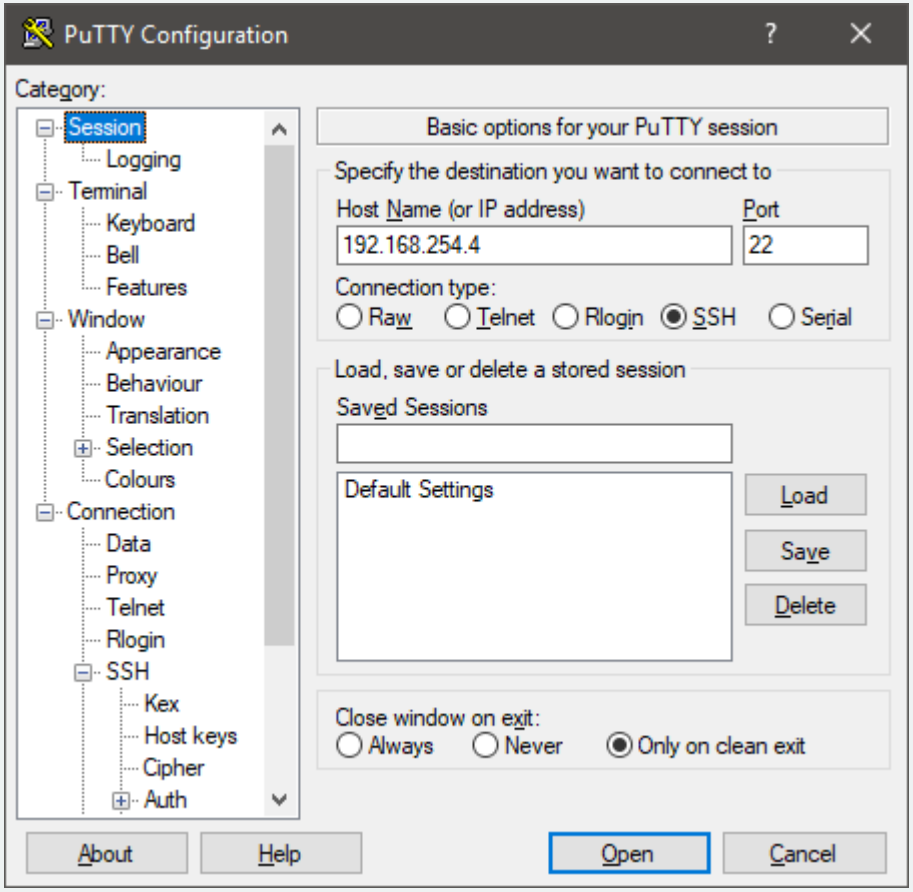
Оконный менеджер

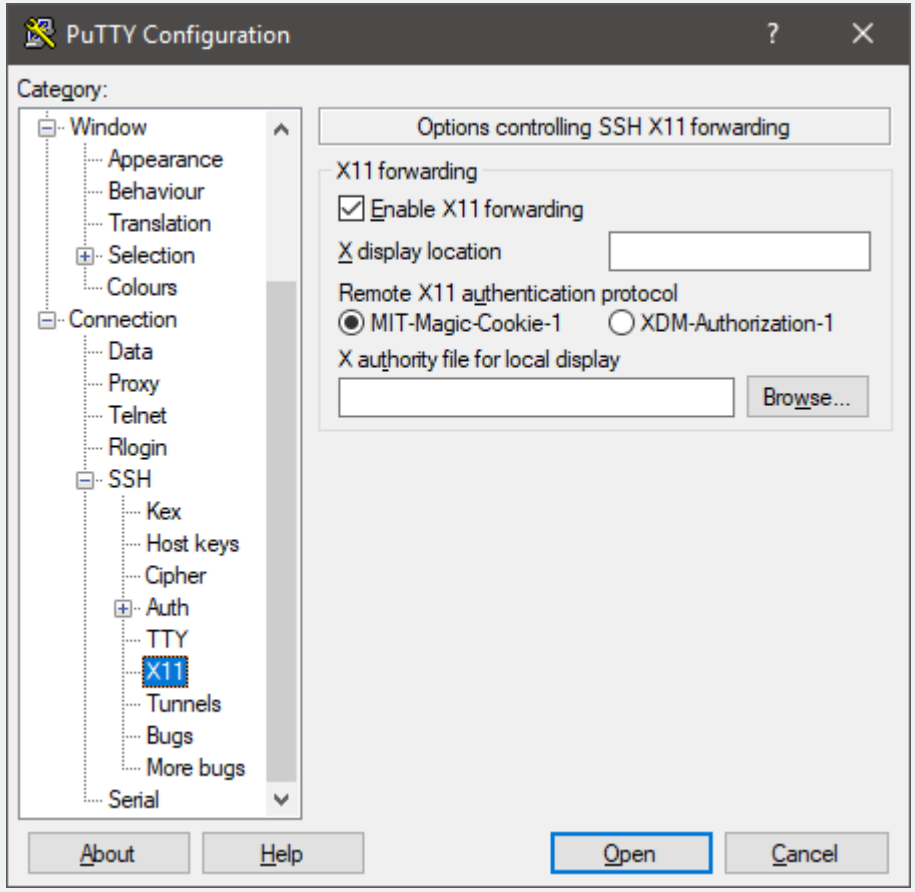


Конечные пользователи



Нелокальное исполнение программ (2)





Нелокальное исполнение программ (3)