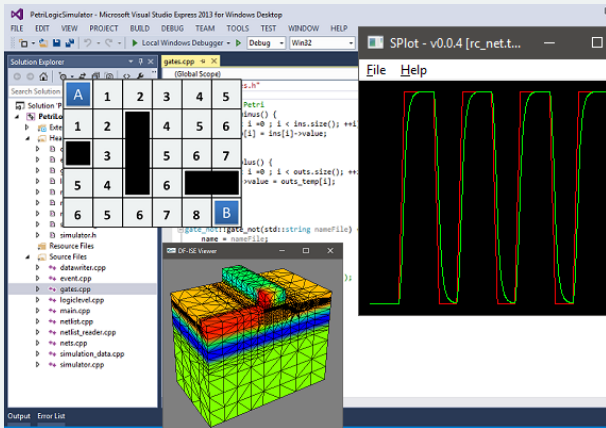




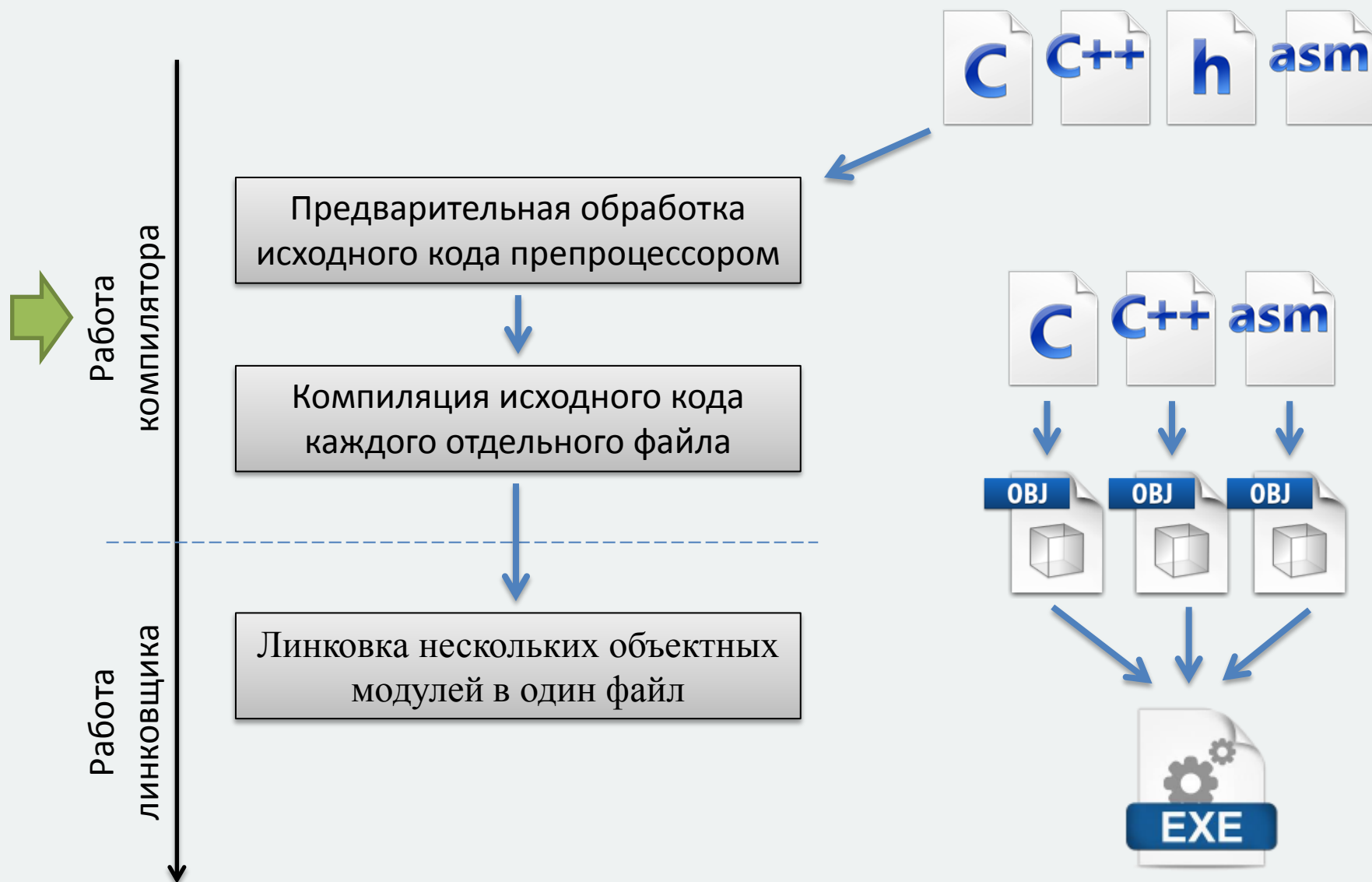
Программные средства САПР



Лекция 2

Этапы сборки программного обеспечения.

Процесс сборки программ



Директивы препроцессора

Подключение заголовочных файлов

```
#include <stdio.h>
#include "myfile.h"
```

Определение значений величин

```
#define N 10

int main() {
    int mas[N], i = 0;

    for (i = 0; i < N; i += 1)
        mas[i] = i;

    return 0;
}
```

Условная компиляция

```
#define N 10
```

→ #define DEBUG_PRINT

```
int main() {
    int mas[N], i = 0;

    for (i = 0; i < N; i += 1)
        mas[i] = i;

    → {
        #if defined(DEBUG_PRINT)
            for (i = 0; i < N; i += 1)
                printf("%d\n", mas[i]);
        #endif

        return 0;
    }
```

Директивы препроцессора: возможные опасности (1)

```
#include <stdio.h>

#define DOUBLE(arg) 2*arg

int main() {
    int a = 4;
    int b = DOUBLE(a);

    printf("%d", b);

    return 0;
}
```

```
#define DOUBLE(arg) 2*(arg)


#include <stdio.h>

#define DOUBLE(arg) 2*arg

int main() {
    int a = 4;
    int b = DOUBLE(a - 1);

    printf("%d", b);

    return 0;
}
```



Директивы препроцессора: возможные опасности (2)

```
#define PI 3.1415
```

```
int main() {  
    float PI = 3.14;  
  
    return 0;  
}
```



main.cpp(4) : warning C4091:

" : ignored on left of 'float' when no variable is declared

main.cpp(4) : error C2143:

syntax error : missing ';' before 'constant'



source.cpp(4): error C2059: syntax error: 'constant'

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

```
#define INC(arg) arg++
```

```
int main() {  
    int a = 4;  
    int b = INC(a - 3);  
}
```



main.cpp(11) : error C2105: '++' needs l-value



Директивы препроцессора: возможные опасности (3)

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

```
#define N 10
```

```
int main() {  
    for (int i = 0; i < N; i++) {  
        printf("Hello\n");  
    }  
    return 0;  
}
```

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

```
#define N 10;
```

```
int main() {  
    for (int i = 0; i < N; i++) {  
        printf("Hello\n");  
    }  
    return 0;  
}
```

Compiling...

main.cpp

source.cpp(6): error C2146: syntax error: missing ')' before identifier 'i'

source.cpp(6): error C2059: syntax error: ';'

source.cpp(6): error C2059: syntax error: '('

source.cpp(7): error C2059: syntax error: ';'

source.cpp(10): error C2059: syntax error: '}'

source.cpp(10): error C2143: syntax error: missing ';' before '}'

Директивы препроцессора: современные возможности

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

```
#include <omp.h>
```

```
#define SIZE 1000
```

```
int main() {
```

```
    printf("%d\n", omp_get_num_procs());
```

```
    omp_set_num_threads(4);
```

```
    int i = 0;
```

```
    int a[SIZE] = { 0 }, b[SIZE] = { 0 }, c[SIZE] = { 0 };
```

```
→ #pragma omp parallel
```

```
{
```

```
→ #pragma omp for
```

```
    for (i = 0; i < SIZE; ++i)
```

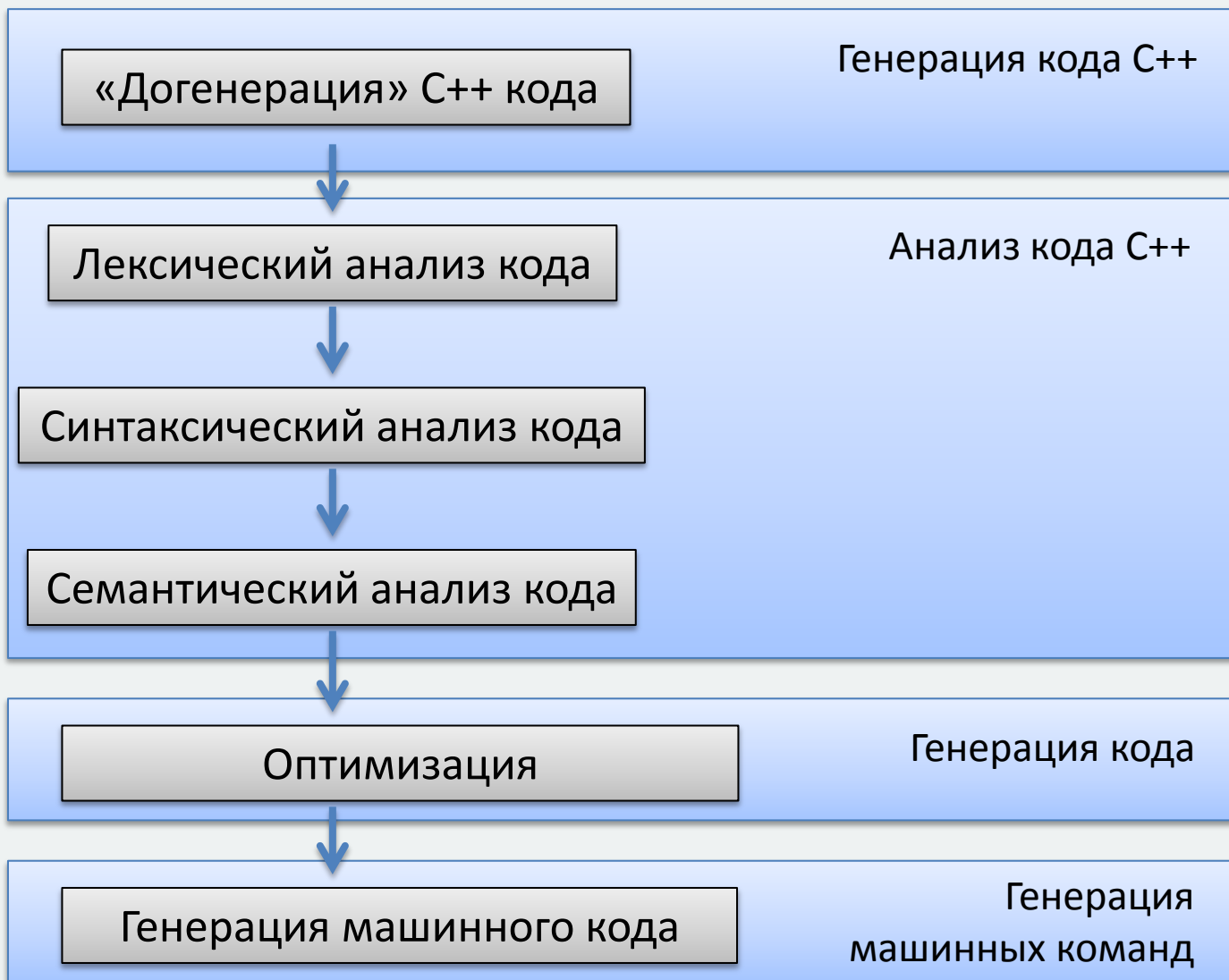
```
        c[i] = a[i] + b[i];
```

```
}
```

```
return 0;
```

```
}
```

Что делает компилятор



«Догенерация» программного кода (1)

```
class SomeClass {  
public:  
    SomeClass() {  
    }  
    ~SomeClass() {  
    }  
};
```

Класс
(конструктор и
деструктор)



Шаблоны



```
template <typename T>  
T test_abs(T arg) {  
    if (arg > 0)  
        return arg;  
    return (-arg);  
}
```

Аргументы
функций по
умолчанию



```
int test_arg(int a, int b = 0) {  
    return a * b;  
}
```


«Догенерация» программного кода (2)

```
int main() {  
    SomeClass sc;  
  
    int x = test_abs(-42);  
    double y = test_abs(-42.0);  
  
    int z = test_arg(x);  
  
    return 0;  
}
```


Используем класс



Вызываем шаблонные
функции с разными
типами аргументов



Вызываем функцию с
одним аргументом





«Догенерация» программного кода: классы

```
SomeClass sc;
```

```
00DF143D lea     ecx,[ebp-11h]
00DF1440 call   SomeClass::SomeClass (0DF100Fh)
00DF1445 mov     dword ptr [ebp-4],0
```

```
int z = test_arg(4);
```

```
00DF144C push   0
00DF144E push   4
00DF1450 call   test_arg (0DF102Dh)
00DF1455 add     esp,8
00DF1458 mov     dword ptr [ebp-20h],eax
```

```
return 0;
```

```
00DF145B mov     dword ptr [ebp-0ECh],0
00DF1465 mov     dword ptr [ebp-4],0FFFFFFFFh
00DF146C lea     ecx,[ebp-11h]
00DF146F call   SomeClass::~~SomeClass (0DF10D2h)
00DF1474 mov     eax,dword ptr [ebp-0ECh]
```



«Догенерация» программного кода: аргументы по-умолчанию

```
int test_arg(int a, int b = 0) {  
    return a * b;  
}
```

```
int main() {  
    int x = 42;
```

```
    int z = test_arg(x);
```

```
    return 0;
```

```
}
```

```
    int z = test_arg(x);  
009C1405  push      0  
009C1407  mov       eax,dword ptr [x]  
009C140A  push      eax  
009C140B  call     test_arg (9C1028h)  
009C1410  add      esp,8  
009C1413  mov     dword ptr [z],eax
```



«Догенерация» программного кода: шаблонные функции

```
template <typename T>
T test_abs(T arg) {
    if (arg > 0)
        return arg;
    return (-arg);
}
```

```
int main() {
    int x = test_abs(-42);
    double y = test_abs(-42.0);

    return 0;
}
```

```
    int x = test_abs(-42);
012013CE  push        0FFFFFFD6h
012013D0  call       test_abs<int> (1201078h)
012013D5  add        esp,4
012013D8  mov        dword ptr [x],eax
    double y = test_abs(-42.0);
012013DB  sub        esp,8
012013DE  fld        qword ptr [__real@c045000000000000
                                (1205740h)]
012013E4  fstp       qword ptr [esp]
012013E7  call       test_abs<double> (12011AEh)
012013EC  add        esp,8
012013EF  fstp       qword ptr [y]
```

Лексический анализ

```
01 : for
02 : (
03 : i
04 : =
05 : 0
06 : ;
07 : i
08 : <
09 : max
10 : ;
11 : +
12 : +
13 : i
14 : )
15 : printf
16 : (
17 : "
18 : Hello
19 : \
20 : n
21 : "
22 : )
23 : ;
```

← for (i = 0; i < max; ++i)
printf("Hello\n");

→ 11 : + → 11 : ++
12 : +

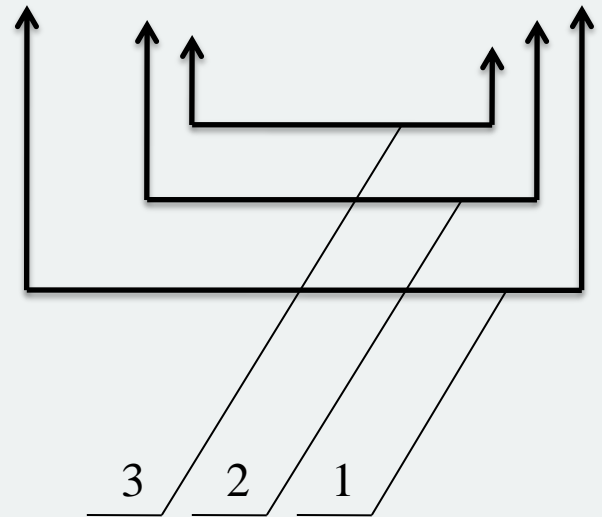
→ 19 : \ → 19 : \n
20 : n

Синтаксический анализ

```
for (i = 0; i < max; ++i)  
    printf("Hello\n");
```

for (A ; B ; C)

```
printf ( D ) ;  
printf ( " Hello\n " ) ;
```





Семантический анализ

```
int i = 0;  
unsigned int max = 10;  
for (i = 0; i < max; ++i)
```

main.cpp(7) : warning C4018: '<' : signed/unsigned mismatch

```
double a = 1.0;  
int b = a;
```

main.cpp(6) : warning C4244: 'initializing' : conversion from 'double' to 'int', possible loss of data

```
int a = sqrt(4);
```

main.cpp(6) : error C2668: 'sqrt' : ambiguous call to overloaded function

1> ..\include\math.h(581): could be 'long double sqrt(long double)'

1> ..\include\math.h(533): or 'float sqrt(float)'

1> ..\include\math.h(128): or 'double sqrt(double)'



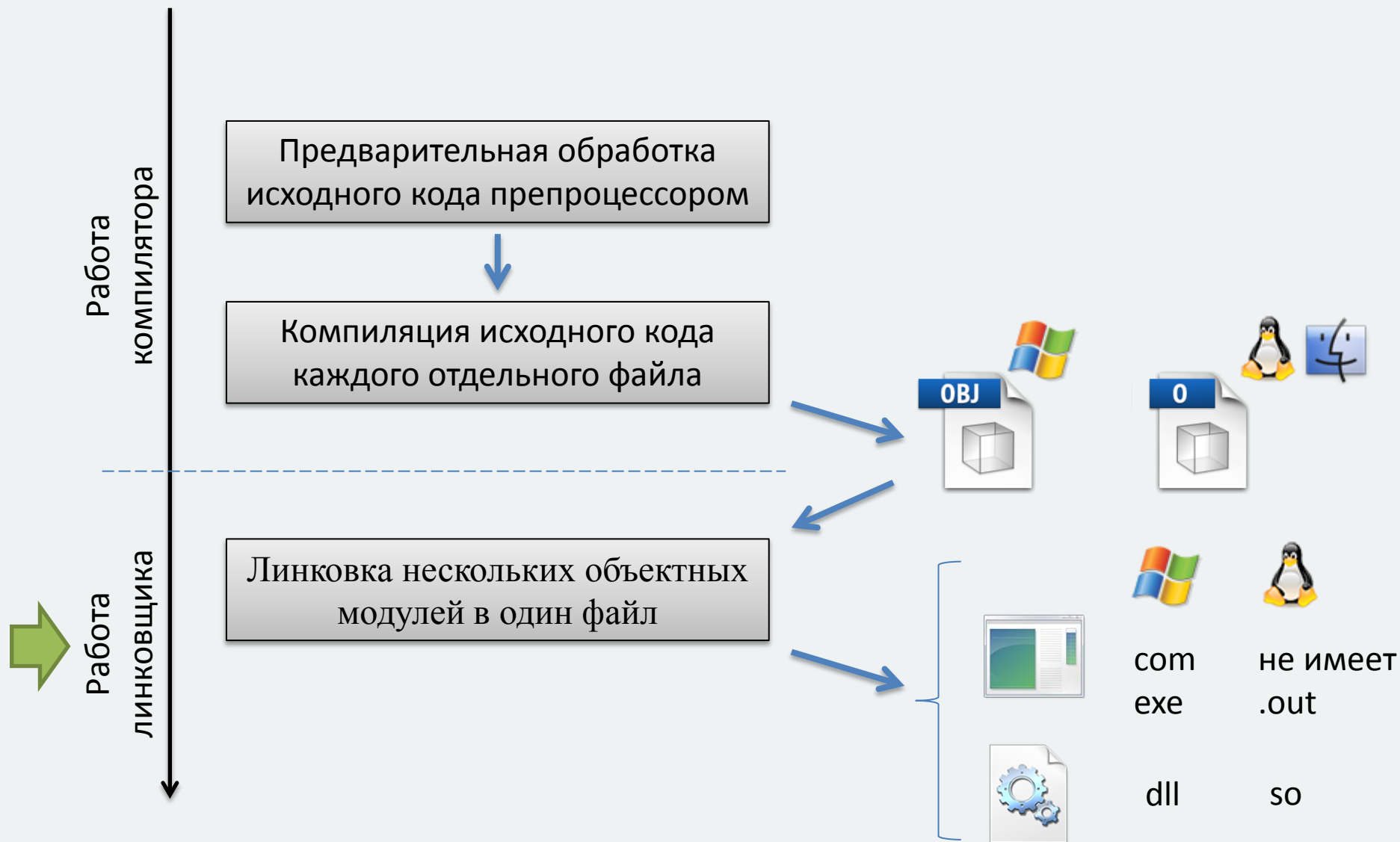
Оптимизация программного кода

Оптимизация выполняется для увеличения производительности

Примеры оптимизаций кода:

1. оптимизация программных инструкций – замена, переупорядочивание;
2. понижение силы операций;
3. оптимизация сложных множественных ветвлений;
4. встраивание inline функций;
5. удаление неиспользуемого кода;
6. выбор регистров;
7. ...

Процесс сборки программ



Что делает компоновщик?

Файл 1.cpp

Локальные для
файла 1
переменные

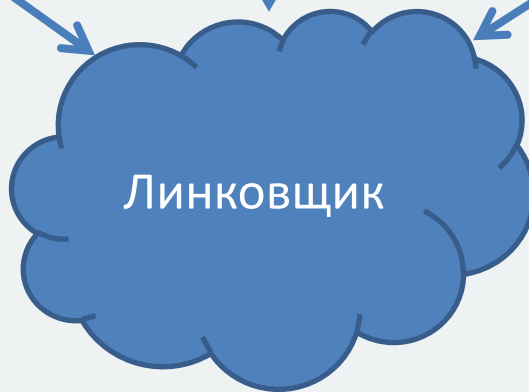
Файл 2.cpp

Локальные для
файла 2
переменные

Файл 3.cpp

Локальные для
файла 3
переменные

Проверка того, что имена
типов данных и
переменных не
пересекаются



Проверка того, что тела
всех вызываемых
функций найдены

Исполняемый
модуль



Ошибки компоновщика (1)

Файл main.cpp:

```
int a = 6;  
  
int main() {  
    int b = 4 + a;  
}
```

Файл func.cpp:

```
int a = 4;  
  
int func() {  
    int c = 2 * a;  
}
```

```
1>main2.obj : error LNK2005: "int a" (?a@@@3HA) already  
defined in main.obj
```



Ошибки компоновщика (2)

Файл main.cpp:

```
#include "header.h"
```

```
int func(int var);
```

```
int main() {  
    func(4);  
}
```

main.obj : error LNK2001: unresolved external symbol "int __cdecl func(int)" (?func@@YAHH@Z)