

Методы выявления и предотвращения недекларированного выполнения программ.

Р.И.Компаниец
В.В.Ковалев

Методология проведения контроля на НДС

Основы методологии

- Общие принципы анализа программ
- Требования РД ФСТЭК России

Базируется на сопоставлении результатов статического анализа исходных текстов ПО и результатов выполнения откомпилированного ПО в процессе динамического анализа

Вопросы для обсуждения

- Метод динамического контроля хода выполнения программ
- Метод статического анализа на основе определяющих отношений подобия
- Обзор инструментов технологии Ирида

Метод динамического контроля

НДВ с позиций теории алгоритмов

| Объект | Запись алгоритма | Вход алгоритма | Выход алгоритма |
|-------------------------|---|--|--------------------------------|
| Алгоритм исходный | Множество операторов $\langle S_1, S_2, \dots, S_N \rangle$ | Операнд $\langle O \rangle$ | Результат $\langle R \rangle$ |
| Алгоритм выполнения | Протокол выполнения (аксиома протокола) | Алгооперанд $\langle O, S \rangle$ в виде связи $r(O, S_1, S_2, \dots, S_N)$ | Алгоритмический процесс |
| НДВ (Алгоритм исходный) | Множество операторов $\langle S_1, S_2, \dots, S_M, \dots, S_N \rangle$ | Операнд $\langle O' \rangle$ | Результат $\langle R' \rangle$ |
| Алгоритм выполнения | Протокол выполнения' | $r'(O', S_1, S_2, \dots, S_M, \dots, S_N)$ | Алгоритмический процесс' |

Метод динамического контроля

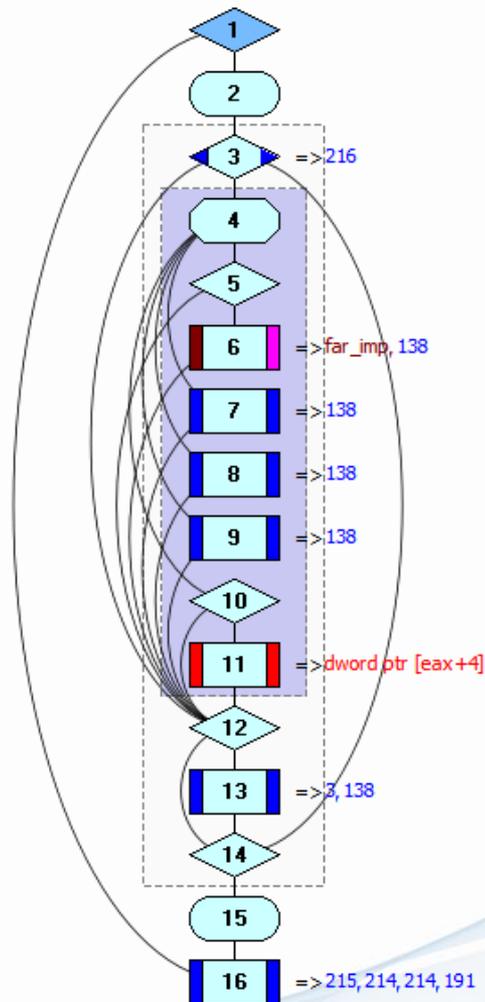
Основные модели

| Модель | Объект описания | Формализм |
|---|---|--|
| Модель алгоритма исходного | Множество операторов $\langle S_1, S_2, \dots, S_N \rangle$, операнд $\langle O \rangle$, результат $\langle R \rangle$ | Управляющий граф программы (УГП). |
| Модель алгоритма выполнения исходного алгоритма (программы) | Множество путей УГП | Порождающая грамматика языка протоколов |
| Модель выявления недокументированного выполнения исходной программы | Множество разрешенных протоколов выполнения исходного алгоритма | Распознающий LL(1) анализатор языка протоколов |

Метод динамического контроля

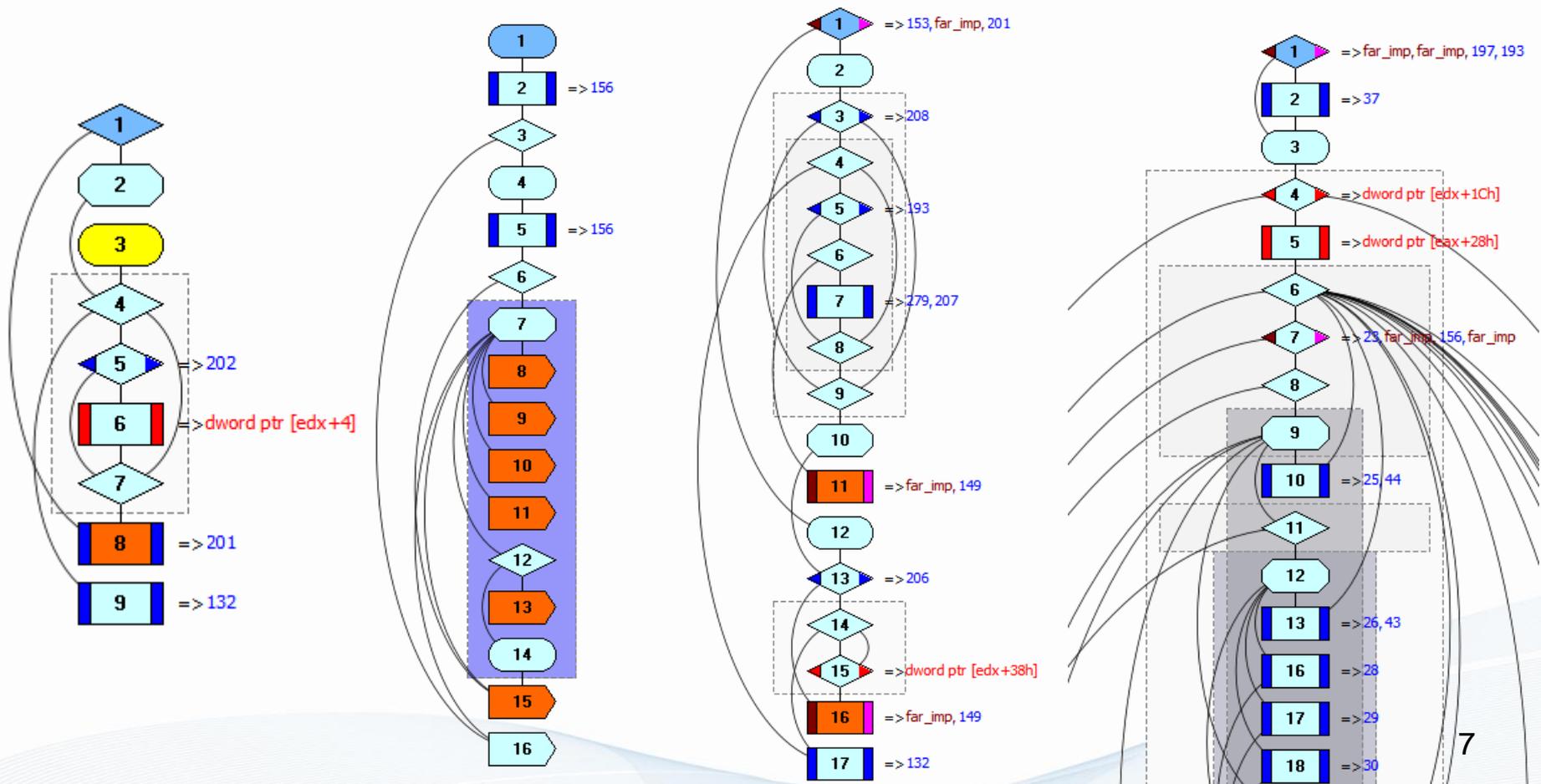
УГП - модели, управляющие структуры

```
void CXRecordset::FreeMapInfoCols ()
{
    POSITION CurPos;
    void * pVoid, * pValue;
    for(CurPos=m_oMapMemElements.GetStartPosition();CurPos!=NULL;)
    {
        m_oMapMemElements.GetNextAssoc (CurPos,pVoid,pValue);
        switch(((TFldInfo *)pVoid)->nFldType)
        {
            case FTString:
            case FTMemo:
                delete (CString*)pValue;
                break;
            case FTInteger:
                delete (int*)pValue;
                break;
            case FTLong:
            case FTCounter:
                delete (long int*)pValue;
                break;
            case FTSingle:
                delete (float*)pValue;
                break;
            case FTDouble:
                delete (double*)pValue;
                break;
            case FTBool:
                delete (BOOL*)pValue;
                break;
            case FTBLOB:
                delete (CByteArray*)pValue;
                break;
        };
        delete (TFldInfo *)pVoid;
    };
    m_oMapMemElements.RemoveAll ();
    oInfoColList.RemoveAll ();
    oValueColList.RemoveAll ();
    oColList.RemoveAll ();
}
```



Метод динамического контроля

Примеры различных УГП



Метод динамического контроля

Представления УГП

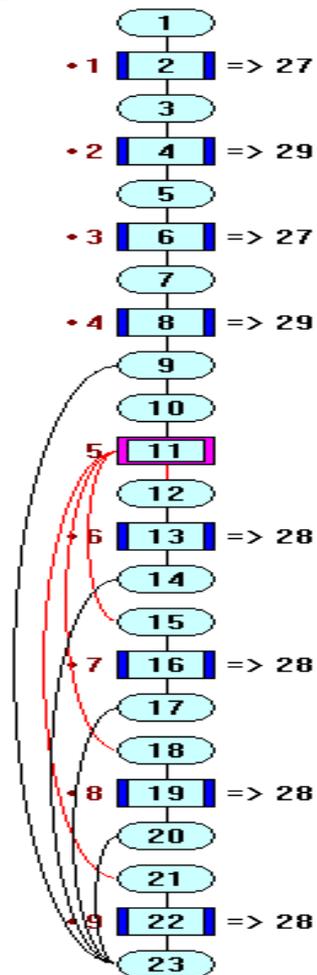
Описание в грамматике

```
//Класс
PassportParser
m0 : m1;
//sub_401000

m1 :
//sub_401000

KT1 m27 KT1
KT2 m29 KT2
KT3 m27 KT3
KT4 m29 KT4
( KT6 m28 KT6 |
KT7 m28 KT7 |
KT8 m28 KT8 |
( KT9 m28 KT9 )?
);
m27 : ;
//sub_401AA0
m28 : ; //sub_401AC0
m29 : ; //sub_401AE0
```

Представление УГП



Пути (протоколы) выполнения

KT1 KT1 KT2 KT2 KT3 KT3 KT4 KT4 ;
KT1 KT1 KT2 KT2 KT3 KT3 KT4 KT4
KT6 KT6 ;
KT1 KT1 KT2 KT2 KT3 KT3 KT4 KT4
KT7 KT7 ;
KT1 KT1 KT2 KT2 KT3 KT3 KT4 KT4
KT8 KT8 ;
KT1 KT1 KT2 KT2 KT3 KT3 KT4 KT4
KT9 KT9.

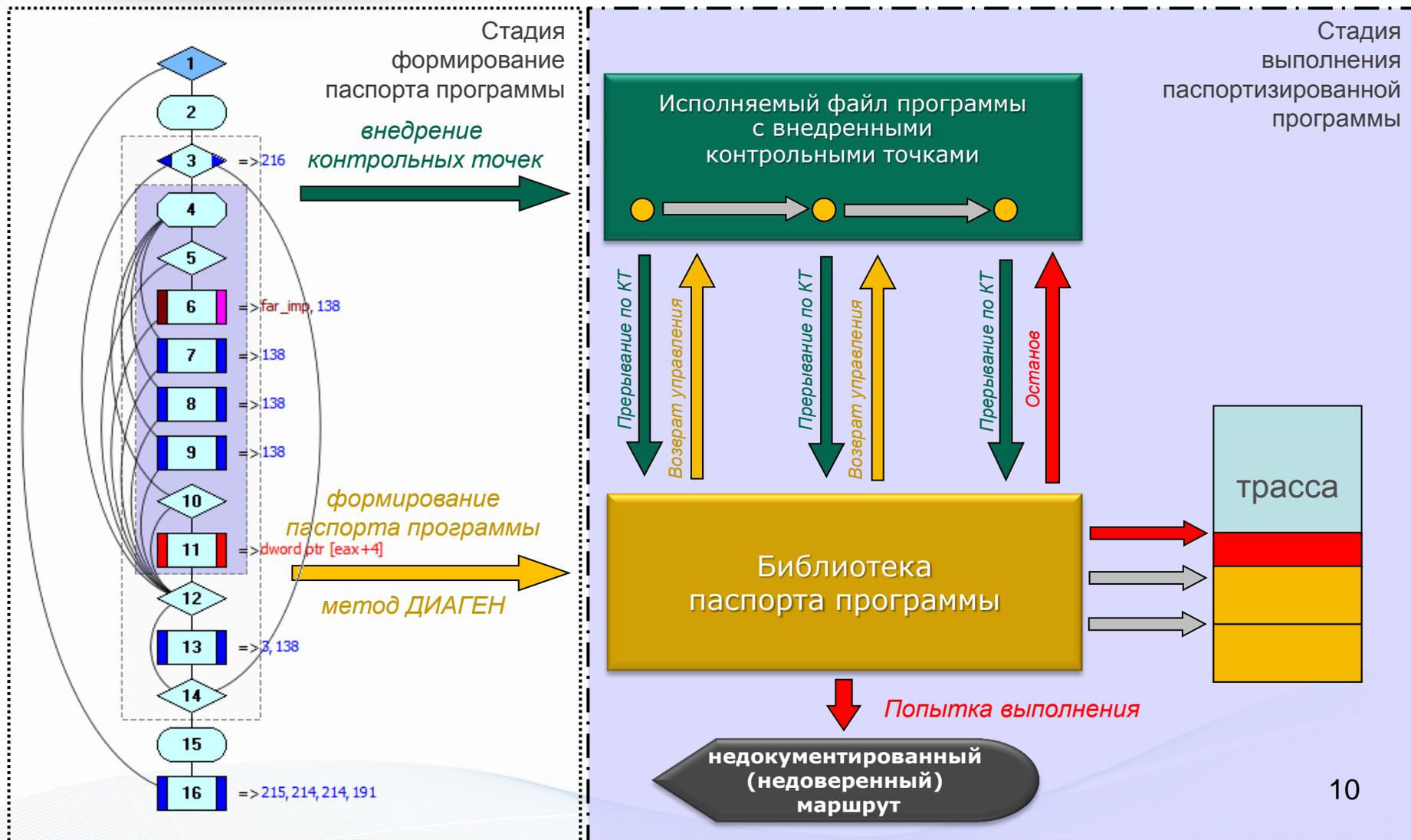
Метод динамического контроля

Основные этапы

| Этап | Исходные данные | Выходные данные | Инструменты |
|--------------------------------------|--|--|-------------------------------------|
| Статический анализ | Исполняемые коды программы | Дизассемблированный код, база данных, УГП. | Дизассемблер IDA PRO, ИК Ирида |
| Паспортизация процессов | УГП Адреса КТ | Паспорт процесса в виде LL(1)-грамматики | ИК Ирида |
| Синтез модуля динамического контроля | Исполняемый код программы Паспорт программы Адреса КТ | Модуль динамического контроля, исполняемый код программы с установленными КТ | ИК Ирида, Компилятор C++; ANTLR.jar |
| Динамический анализ и контроль | Исполняемый код программы с установленными КТ Модуль динамического контроля | Протокол выполнения программы | ИК Ирида |

Метод динамического контроля

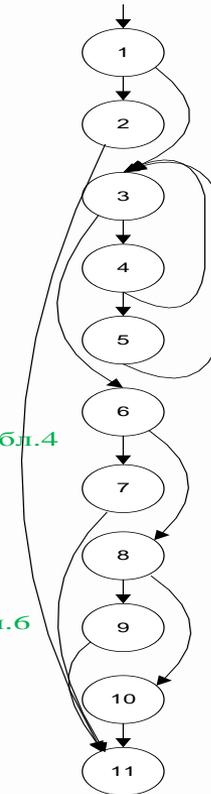
Механизм работы метода



Статический анализ на основе отношений подобия

Пример программы с НДВ и ее УГП

```
int _tmain(int argc, _TCHAR* argv[])
{
    string passwd_file = "password.txt";
    ifstream passwd_stream( passwd_file );
    if ( ! passwd_stream.is_open() )//Нижняя гр. бл.1
    {
        string err = strerror( errno );
        cerr << "Cannot open file " << passwd_file << " : "
            << errno << " : " << err << endl;
        exit( -1 );//Нижняя гр. бл.2
    }
    string line;
    while( getline( passwd_stream, line ) )//Нижняя гр. бл.3
    {
        istringstream iss(line);
        string name;
        string password;
        if ( !(iss >> name >> password) ) continue;//Нижняя гр. бл.4
        passwords[name] = password;//Нижняя гр. бл.5
    }
    string name, password;
    cout << "Enter name:"; cin >> name;
    cout << "Enter password:"; cin >> password;
    if ( name == "admin" && password == "111" ){//Нижняя гр. бл.6
        cout << "Welcome!" << endl; //Нижняя гр. бл.7
    }
    else
    {
        if (passwords.find(name)!= passwords.end()
            &&passwords[name]==password) //Нижняя гр. бл.8
            cout << "Welcome!" << endl;//Нижняя гр. бл.9
        else
            cout << "Access denied!" << endl;//Нижняя гр. бл.10
    }
    return 0;//Нижняя гр. бл.11
}
```



Статический анализ на основе отношений подобия

Основные этапы

- Построение УГП
- Формирование абстрактных размерностей векторов входных и выходных данных
- Построение определяющих отношений
- Выделение артикляционных компонент программы (АКП)
- Проверка совместимости определяющих отношений подобия (ОО)

Статический анализ на основе отношений подобия

Техника формирования ОО

Единственное ОО

$$\left\{ \begin{array}{l} a = (v_t - v_0)/t, \\ s = v_0 t + at^2/2. \\ s = v_0 t + ((v_t - v_0)/t)t^2/2 = v_0 t + v_t t - v_0 t. \\ [v_0 t] = [v_t t] \text{ или } [v_0] = [v_t] \\ \prod_{k=1}^r [x_k]^{a_k} = 1 \text{ или } \prod_{k=1}^r [x_k]^{a_k} = [.]^0 \\ [v_0] [v_t]^{-1} = [.]^0. \end{array} \right.$$

$$\sum_{k=1}^r a_k \ln[x_k] = 0, \xrightarrow{\ln[x_k]=z_k} \sum_{k=1}^r a_k z_k = 0 \longrightarrow Az = 0$$

Внесение ошибки (замена / на *) - моделирование НДВ (X и Y те же)

Три ОО

$$a = (v_t - v_0)t \xrightarrow{\text{sup}} s = v_0 t + ((v_t - v_0)t)t^2/2 = v_0 t + v_t t^3 + v_0 t^3$$

$$\left\{ \begin{array}{l} [v_0 t] = [v_t t^3] \quad [v_0] [v_t]^{-1} [t]^2 = [.]^0 \\ [v_0 t] = [v_0 t^3] \longrightarrow [t]^2 = [.]^0 \\ [v_0 t^3] = [v_t t^3] \quad [v_0] [v_t]^{-1} = [.]^0 \end{array} \right. \xrightarrow{[X] = ([v_0], [v_t], [t])} A = \begin{pmatrix} 1 & -1 & -2 \\ 0 & 0 & -2 \\ 1 & -1 & 0 \end{pmatrix} \xrightarrow{[Y] = ([a], [s])}$$

Система ЛОУ несовместна $|X|=3$

t-безразмерная

Статический анализ на основе отношений подобия

Формирование абстрактных размерностей

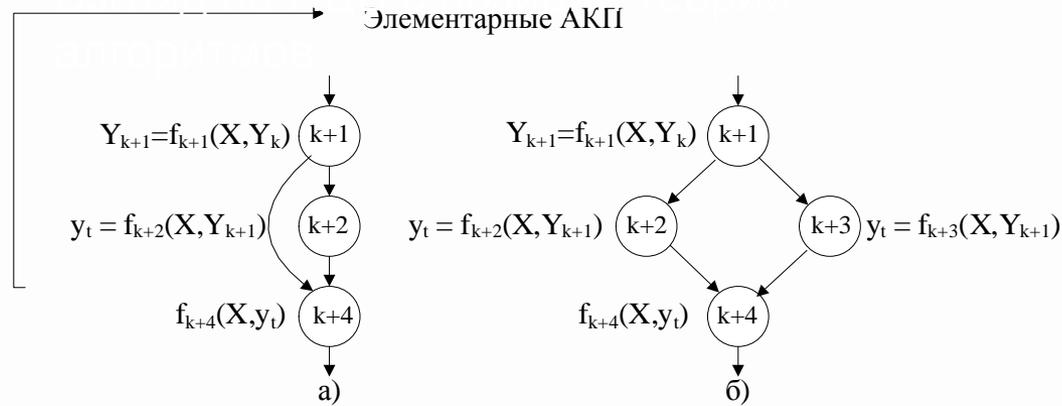
$Y = F(X, Y)$, $X = \{x_i\}$ – исх.данные, $Y = \{y_j\}$ – результаты (имена)

X и Y – упорядочиваются и отображаются на вектора X и Y

Вектор $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$; Вектор $Y = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$;

"password.txt"
erno
passwd_stream
cin
"admin"
"111"
"Welcome!"
"Access denied!"

passwd_file
err
line
iss
name
password
passwords[name]
cout



$T : X \rightarrow [X]$

для а) : $[f_{k+4}(X, f_{k+1}(X, Y_k))] = [f_{k+4}(X, f_{k+2}(X, f_{k+1}(X, Y_k)))]$, (1)

для б) : $[f_{k+4}(X, f_{k+2}(X, Y_{k+1}))] = [f_{k+4}(X, f_{k+3}(X, Y_{k+1}))]$. (2)

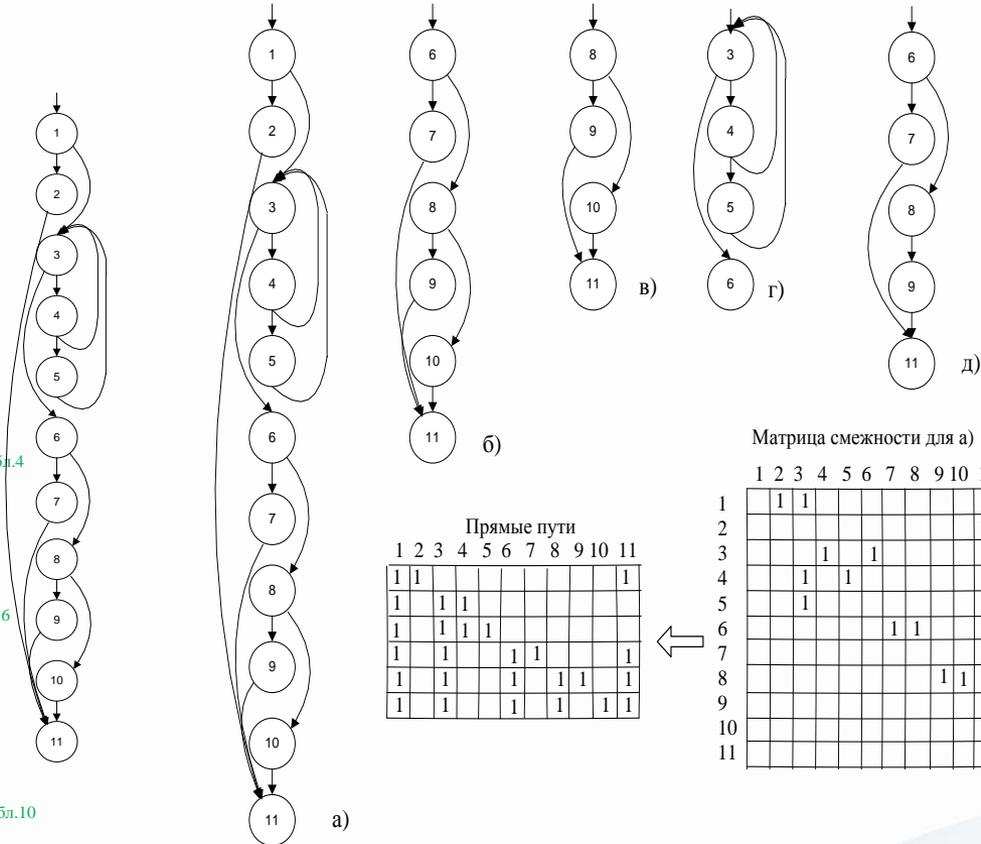
Статический анализ на основе отношений подобия

Разложение УГП на артикуляционные комплексы

```

int _tmain(int argc, _TCHAR* argv[])
{
    string passwd_file = "password.txt";
    ifstream passwd_stream( passwd_file );
    if ( ! passwd_stream.is_open() )//Нижняя гр. бл.1
    {
        string err = sterror( errno );
        cerr << "Cannot open file " << passwd_file << " : "
            << errno << " : " << err << endl;
        exit( -1 );//Нижняя гр. бл.2
    }
    string line;
    while( getline( passwd_stream, line ) )//Нижняя гр. бл.3
    {
        istream iss(line);
        string name;
        string password;
        if ( !(iss >> name >> password) ) continue;//Нижняя гр. бл.4
        passwords[name] = password;//Нижняя гр. бл.5
    }
    string name, password;
    cout << "Enter name:";   cin >> name;
    cout << "Enter password:"; cin >> password;
    if ( name == "admin" && password == "111" ){//Нижняя гр. бл.6
        cout << "Welcome!" << endl; //Нижняя гр. бл.7
    }
    else
    {
        if (passwords.find(name)!= passwords.end()
            && passwords[name]!=password) //Нижняя гр. бл.8
            cout << "Welcome!" << endl; //Нижняя гр. бл.9
        else
            cout << "Access denied!" << endl; //Нижняя гр. бл.10
    }
    return 0; //Нижняя гр. бл.11
}

```



Прямые пути

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|----|---|---|---|---|---|---|---|---|---|----|----|
| 1 | 1 | 1 | | | | | | | | | 1 |
| 2 | 1 | 1 | 1 | | | | | | | | |
| 3 | | | | 1 | 1 | | | | | | |
| 4 | | | | 1 | 1 | | | | | | |
| 5 | | | | 1 | | | | | | | |
| 6 | | | | | | | | | | | |
| 7 | | | | | | | 1 | 1 | | | |
| 8 | | | | | | | 1 | 1 | 1 | 1 | |
| 9 | | | | | | | 1 | 1 | 1 | 1 | |
| 10 | | | | | | | | | | | 1 |
| 11 | | | | | | | | | | | 1 |

Матрица смежности для а)

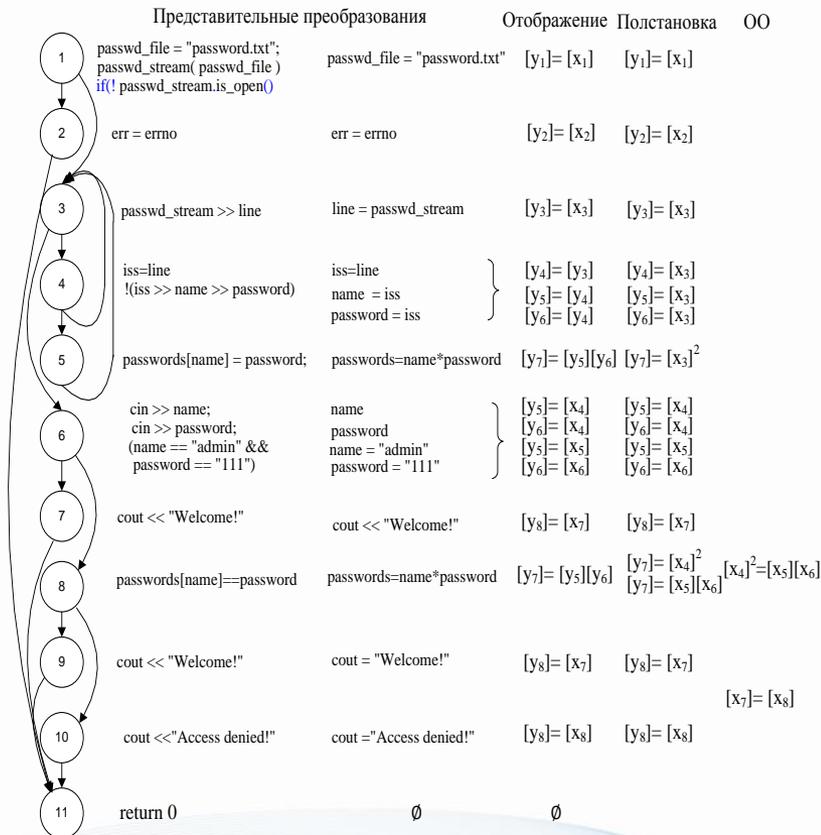
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|----|---|---|---|---|---|---|---|---|---|----|----|
| 1 | | 1 | 1 | | | | | | | | |
| 2 | | | | | | | | | | | 1 |
| 3 | | | | 1 | 1 | | | | | | |
| 4 | | | | 1 | 1 | | | | | | |
| 5 | | | | 1 | | | | | | | |
| 6 | | | | | | | | | | | |
| 7 | | | | | | | 1 | 1 | | | |
| 8 | | | | | | | 1 | 1 | 1 | 1 | |
| 9 | | | | | | | | | | | 1 |
| 10 | | | | | | | | | | | 1 |
| 11 | | | | | | | | | | | 1 |

Статический анализ на основе отношений подобия

Формирование и проверка определяющих отношений подобия

Построение определяющих отношений

Проверка совместимости артикуляционных компонент



$$(6, 7, 11): \left\{ \begin{array}{l} [y_5]=[x_4]; \\ [y_6]=[x_4]; \\ [y_5]=[x_5]; \\ [y_6]=[x_6]; \\ [y_8]=[x_7]; \end{array} \right\} \quad (6, 8, 9, 11): \left\{ \begin{array}{l} [y_5]=[x_4]; \\ [y_6]=[x_4]; \\ [y_5]=[x_5]; \\ [y_6]=[x_6]; \\ [y_7]=[x_4] [x_4]; \\ [y_7]=[x_5] [x_6]; \\ [y_8]=[x_7]; \end{array} \right\} \quad (6, 8, 10, 11): \left\{ \begin{array}{l} [y_5]=[x_4]; \\ [y_6]=[x_4]; \\ [y_5]=[x_5]; \\ [y_6]=[x_6]; \\ [y_7]=[x_4] [x_4]; \\ [y_7]=[x_5] [x_6]; \\ [y_8]=[x_8]; \end{array} \right\}$$

$$(6, 7, 11): \left\{ \begin{array}{l} [y_5]=[x_4]; \\ [y_6]=[x_4]; \\ [y_5]=[x_5]; \\ [y_6]=[x_6]; \\ [y_8]=[x_7]; \end{array} \right\} \quad (6, 8, 9, 11): \left\{ \begin{array}{l} [y_5]=[x_4]; \\ [y_6]=[x_4]; \\ [y_5]=[x_5]; \\ [y_6]=[x_6]; \\ [y_7]=[x_4] [x_4]; \\ [y_7]=[x_5] [x_6]; \\ [y_8]=[x_7]; \end{array} \right\}$$

$$(6, 7, 11): \{ [y_7]=[x_3][x_3]; \} \quad (6, 8, 9, 11) = \left\{ \begin{array}{l} [y_7]=[x_4][x_4]; \\ [y_7]=[x_5][x_6]; \end{array} \right\}$$

$$[x_3] = [x_4] - [\text{объект из файла1}] = [\text{объект из файла2}];$$

$$[x_3] = [x_5] - [\text{объект из файла}] = [\text{константа1}];$$

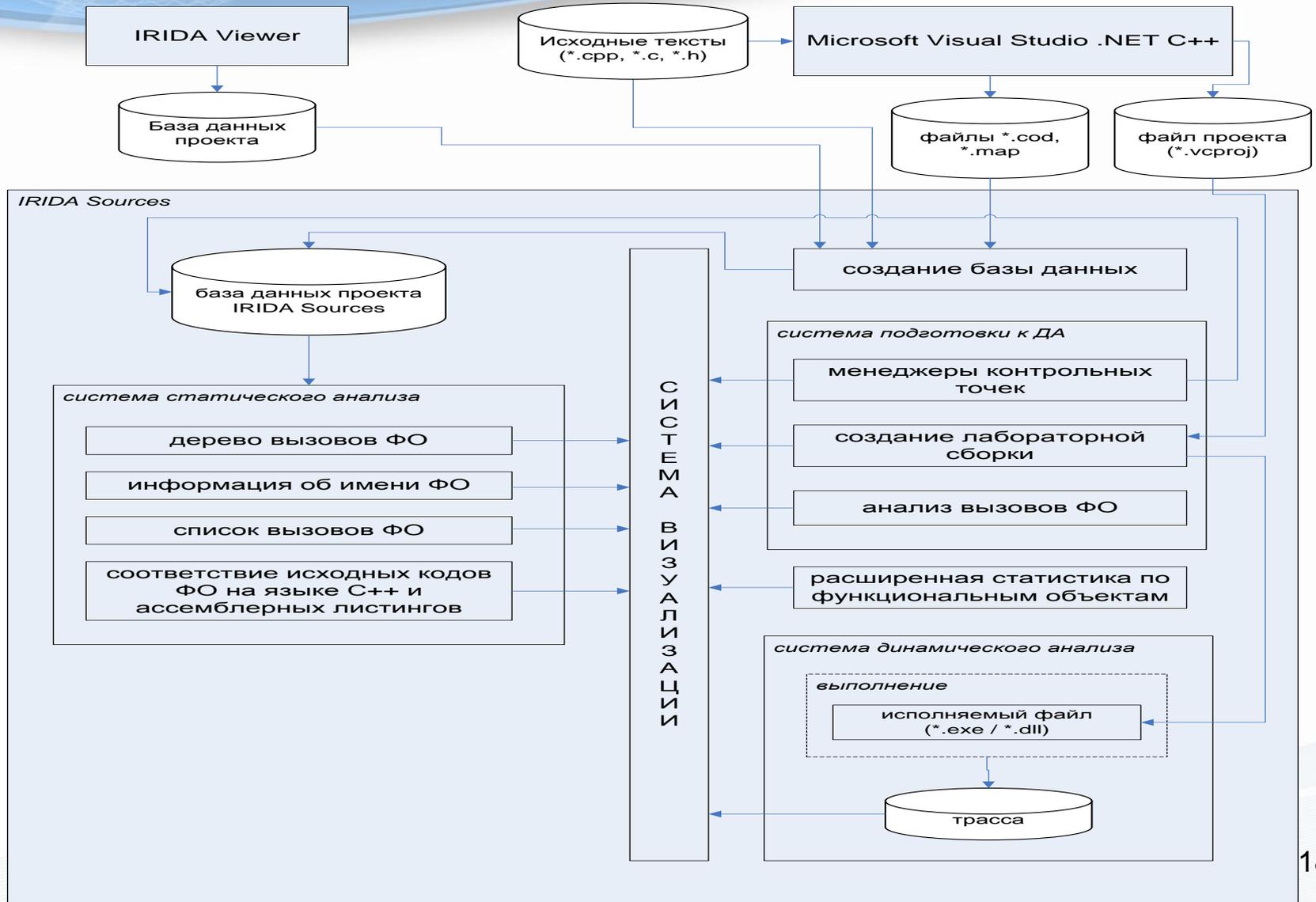
$$[x_3] = [x_6] - [\text{объект из файла}] = [\text{константа2}].$$

$$A = \begin{pmatrix} 2 & -2 & 0 & 0 \\ 0 & 2 & -1 & -1 \\ 2 & 0 & -1 & -1 \end{pmatrix}$$

Инструменты технологии Ирида

| Наименование ИК | Назначение | Целевая платформа |
|-----------------|---|-------------------------------------|
| Irida Sources | Статический и динамический анализ C++ и C программ | Microsoft Visual Studio .NET C++ |
| Irida 1.1 | Статический и динамический анализ, динамический контроль исполняемых кодов программ | Intel x86 |
| Irida 2.0 | Статический и динамический анализ, динамический контроль исполняемых кодов программ | Intel x86, ARM, Power PC и др. |
| | | |

Структура ИК IRIDA Sources



Инструменты IRIDA Sources

The screenshot displays the IRIDA Sources IDE interface. The main window shows the C++ source code for `AnalyseFilesChoiceDlg.cpp`. The code defines the `CAnalyseFilesChoiceDlg` class, including a constructor and a `for` loop that iterates over `ShowFilter` and `AnalyseFilter` arrays. The assembly view below shows the corresponding assembly instructions, such as `mov`, `jmp`, `add`, and `jge`. The call tree on the right shows the function `InsertImageListIcon` calling `AfxGetApp`, `LoadIconA`, and `Add`. The status bar at the bottom indicates the system is ready.

IRIDASources - [IRIDASourcesNew - aaaa]

Файл Правка Анализ Вид Лабораторная сборка Настройка Помощь

AnalyseFilesChoiceDlg.cpp

1 GetBaseClass
GetThisClass
2 GetRuntimeClass
3 CAnalyseFilesChoiceDlg
4 `scalar deleting destructor'
5 ~CAnalyseFilesChoiceDlg
6 InsertImageListIcon
7 DoDataExchange
8 GetMessageMap
9 GetThisMessageMap
10 OnInitDialog
11 FillList
12 OnBnClickedButton3
13 OnNMDblclkList1
14 ChangeRealPath
15 OnBnClickedButton4
16 OnNMCustomdrawList1
17 OnLvnItemchangedList1
18 OnBnClickedCancel
19 UpdateAnalyse
20 Filter
21 OnBnClickedCheckShow
22 OnBnClickedCheckAnalyse
FileStateChoiceDlg.h
23 IsApp
AsmFrame.cpp
TextView.h
ClipboardOperDecor.h

C++ ассемблер

```
CAnalyseFilesChoiceDlg::CAnalyseFilesC  
{  
    ProjFileLst = _FileLst;  
    CancelFlag = false;  
  
    for(int i = 0; i < 2; i++){  
        ShowFilter[i] = true;  
        AnalyseFilter[i] = true;  
    }  
  
    cImageList.Create(16,16,ILC_COLOR4  
    InsertImageListIcon();  
}
```

Дерево вызовов

- 3 CAnalyseFilesChoiceDlg
 - CDialog
 - CListCtrl
 - CImageList
 - Create
 - 6 InsertImageListIcon
 - AfxGetApp
 - LoadIconA
 - Add
 - AfxGetApp
 - LoadIconA
 - Add

Соответствие текстов C - asm

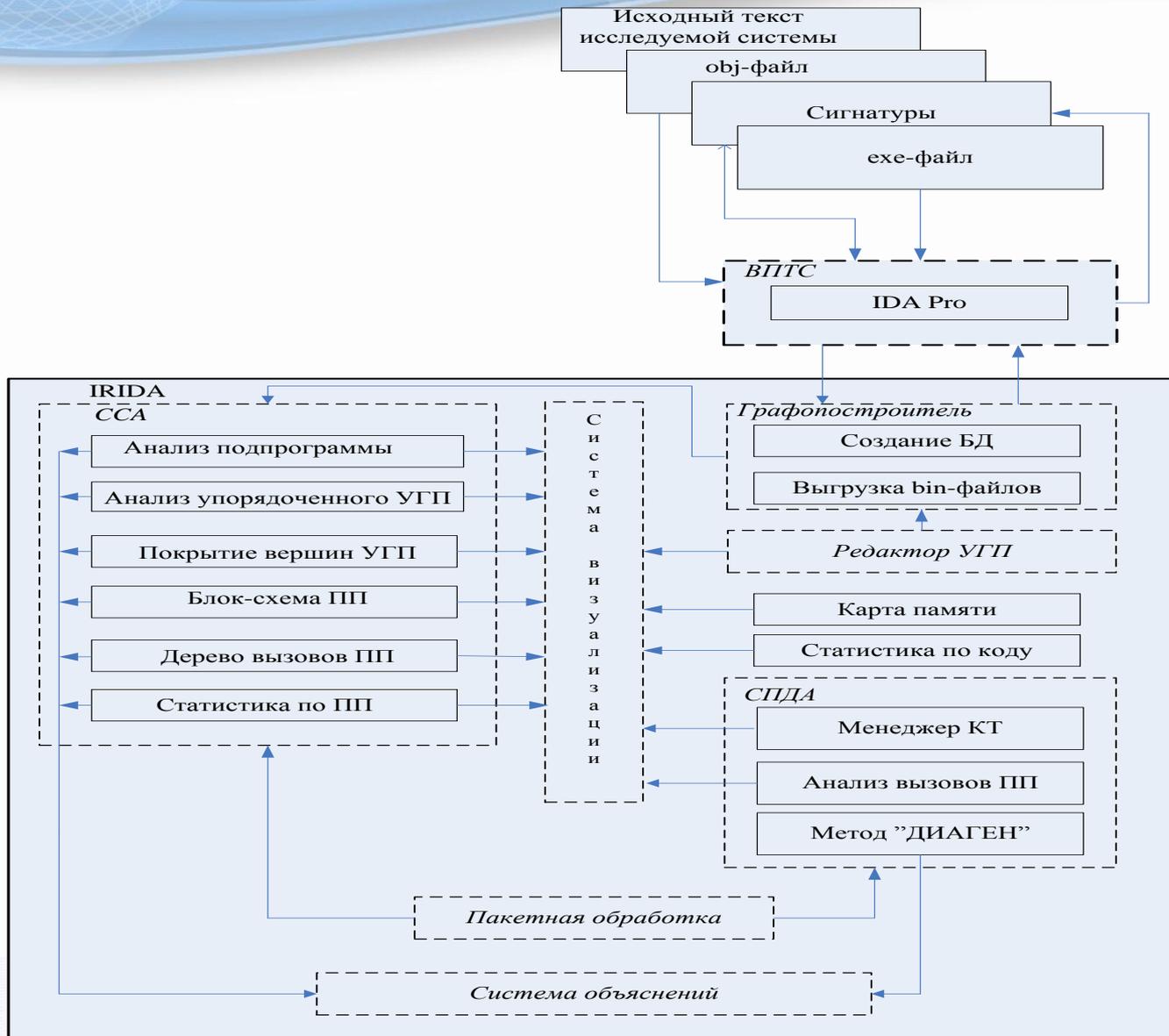
```
CAnalyseFilesChoiceDlg::CAnalyseFilesChoiceDlg(CProjFileL  
{  
    ProjFileLst = _FileLst;  
    CancelFlag = false;  
  
    for(int i = 0; i < 2; i++){  
        ShowFilter[i] = true;  
        AnalyseFilter[i] = true;  
    }  
  
    cImageList.Create(16,16,ILC COLOR4,2,0);  
}
```

```
00083 c7 45 ec 00 00  
00 00 mov DWORD PTR _i@0, 0  
0008a eb 09 jmp SHORT $LN3@CAnalyseFi  
$LN2@CAnalyseFi:  
0008c 8b 45 ec mov eax, DWORD PTR _i@0  
0008f 83 c0 01 add eax, 1  
00092 89 45 ec mov DWORD PTR _i@0, eax  
$LN3@CAnalyseFi:  
00095 83 7d ec 02 cmp DWORD PTR _i@0, 2  
00099 7d 24 jge SHORT $LN1@CAnalyseFi
```

Ready

Пуск IRIDASources - [IRID... EN 23:01

Структура инструментального комплекса IRIDA



Инструментальный комплекс IRIDA

IRIDA 1.1

Проект: Calc_test; Сессия: Session 2 - IRIDA Viewer

Файл Правка Анализ Вид Язык Помощь

Подпрограммы

| N° | Метка п/п-мы |
|----|-------------------------|
| 1 | sub_401000 |
| 2 | _main |
| 3 | sub_4012E0 |
| 4 | ?fail@ios_base@std@@@QB |
| 5 | unknown_libname_1 |
| 6 | sub_401340 |
| 7 | sub_401360 |
| 8 | sub_401380 |
| 9 | sub_4013A0 |
| 10 | sub_4013C0 |
| 11 | sub_4013E0 |
| 12 | sub_401400 |
| 13 | ??1@ios_base@std@@@QB |

Линейные участки

| N° | Метка блока |
|----|-------------|
| 1 | sub_401000 |
| 2 | _1 |
| 3 | _2 |
| 4 | _3 |
| 5 | _4 |
| 6 | _5 |
| 7 | _6 |
| 8 | _7 |
| 9 | _8 |
| 10 | _9 |
| 11 | _10 |
| 21 | loc_40108F |
| 12 | loc_4010A9 |

WinGraph32 - Subroutine: sub_401000

Инструкции

| Адрес | Имя | Оп. 1 | Оп. 2 | Оп. 3 | Стек | N° строки |
|----------|------|----------------|----------------|-------|------|-----------|
| 00401000 | push | ebp | | | 000 | 76 |
| 00401001 | mov | ebp | esp | | 004 | 77 |
| 00401003 | sub | esp | 20h | | 004 | 78 |
| 00401006 | mov | [ebp + var_1C] | ecx | | 024 | 79 |
| 00401009 | mov | ecx | [ebp + var_1C] | | 024 | 80 |

Информация

Тек. п/п-ма: sub_401000
номер: 1
начало: 61
конец: 151

Минимальное покрытие вершин управляющего графа для подпрограммы 'sub_401000'

| N° | Показ. | Точки трассировки | Путь |
|----|-------------------------------------|-------------------------------------|-------------------------------------|
| 1 | | <input type="checkbox"/> | 1 2 3 4 5 6 7 8 9 10 11 21 22 23 |
| 2 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 23 |
| 3 | | <input type="checkbox"/> | 1 2 3 4 5 6 7 8 9 10 11 15 16 17 23 |
| 4 | | <input type="checkbox"/> | 19 20 23 |
| 5 | | <input type="checkbox"/> | 1 2 3 4 5 6 7 8 9 10 11 18 |

Ограничения на установку точек трассировки

Без ограничений Устанавливать только на след. переходы:

- Ближний вызов
- Передача управления на блок с данными
- Дальний вызов
- Передача управления в тело другой подпрограммы
- Регистровая передача управления
- Передача управления в свернутую подпрограмму
- Передача управления путем явной адрессации

Установить и показать

Subroutine 'sub_401000' (1) Call Tree

П/п-мы, вызывающие данную п/п-му

| N° | Метка п/п-мы | Кол-во |
|----|--------------|--------|
| 2 | _main | 1 |

1
• 1 2 => 27
3
• 2 4 => 29
5
• 3 6 => 27
7
• 4 8 => 29
9
10
11
12
• 6 13 => 28
14
15
7 16 => 28
17
8 19 => 28
20
9 22 => 28
18
23

Инструменты IRIDA 2.0

Framework

File View Navigation Settings Static Analysis Dynamic Analysis

HEX M 2 + - IDA-ASM IRIDA-ASM

Segments

| title | addr_beg | addr_end |
|----------|----------|----------|
| 1 .text | 4198400 | 4206592 |
| 2 .idata | 4206592 | 4206880 |
| 3 .rdata | 4206880 | 4214784 |
| 4 .data | 4214784 | 4218880 |

Subs-IRIDA-ASM

| num | title | det |
|-----|------------|-----|
| 1 1 | _main | |
| 2 2 | sub_4011C0 | |
| 3 3 | sub_401250 | |
| 4 4 | sub_4012F0 | |
| 5 5 | sub_401480 | |
| 6 6 | sub_401850 | |

Subs-IRIDA-ASM Subs-IDA-ASM

Instructions-IRIDA-ASM

| id | block_num | num | address | title | OP1 | OP2 | OP3 | OP4 | stack | comments | repeatable_comments | auto_comments |
|----|-----------|-----|---------|-------|-----|------------|-----|-----|-------|----------|---------------------|--------------------------------|
| 1 | 1 | 1 | 4198400 | push | ebp | | | | 0 | | | // Push Operand onto the Stack |
| 2 | 2 | 1 | 4198401 | mov | ebp | esp | | | 4 | | | // Move Data (from to) |
| 3 | 3 | 1 | 4198403 | and | esp | 0FFFFFFF8h | | | 4 | | | // Logical AND |

Sub 1 - _main

Sub 1 - _main

Sub Call Graph 1 - _main

Выводы

- НДВ всегда присутствуют в ПО
- Контроль отсутствия/наличия НДВ в ПО требует специализированных методик и инструментов
- Создание условий невозможности проявления НДВ (запрета выполнения недоверенного кода) реализуемо программно на практике
- Паспортизация ПО – один из путей решения проблемы защиты ПО как от скрытых НДВ, так и от атак на ПО в памяти

Спасибо за внимание

Компаниец Радион Иванович
kompaniec-r@gaz-is.ru

Ковалев Виктор Васильевич
kovalev-v@gaz-is.ru

ООО «Газинформсервис»
г. Санкт-Петербург
www.gaz-is.ru