

Лекция №12

- **ОСНОВЫ ИНТЕГРАЦИИ
с ANSYS: работа в
ИНТЕРАКТИВНОМ И ПАКЕТНОМ
режимах.**

Программный комплекс ANSYS не поддерживает рассмотренные ранее стандартные технологии взаимодействия между приложениями. Основой для интеграции прикладных приложений с программным комплексом является возможность использования входных (Input) и выходных (Output) файлов. Кроме того, может осуществляться сохранение ряда данных в виде текстовых и графических файлов.

При работе с ANSYS в интерактивном режиме, как правило, используется однонаправленная передача данных от пользовательского приложения. В этом случае алгоритм интеграции сводится к следующим трем действиям:

- 1. Подготовка данных.**
- 2. Формирование входного файла с набором команд ANSYS и операторов APDL (ANSYS Parametric Design Language).**
- 3. Запуск ANSYS.**

При этом, входной файл должен быть назван startXX.ans. Где XX – это цифры основной версии и модификации программного комплекса ANSYS.

**Т.е. для ANSYS 5.7 - start57.ans,
для ANSYS 8.0 - start80.ans и т.д.**

Среди параметров запуска ANSYS не должен присутствовать –s poread.

Входной файл должен находиться в рабочей директории.

Запуск ANSYS может производиться при помощи функции:

- **function WinExec(lpCmdLine: LPCSTR; uCmdShow: UINT): UINT; stdcall;**

Функция выполняет запуск командной строки (полное имя приложения + параметры), которая определяется параметром lpCmdLine. Здесь тип LPCSTR соответствует PChar. Переменные и выражения типа String должны явно приводиться к типу PChar.

Например:

```
WinExec (PChar (Edit1.Text) , 1) ;
```

Параметр `uCmdShow` определяет каким должно быть окно запускаемого приложения:

`SW_SHOWNORMAL` – «обычное»;

`SW_SHOWMINIMIZED` – свернутое;

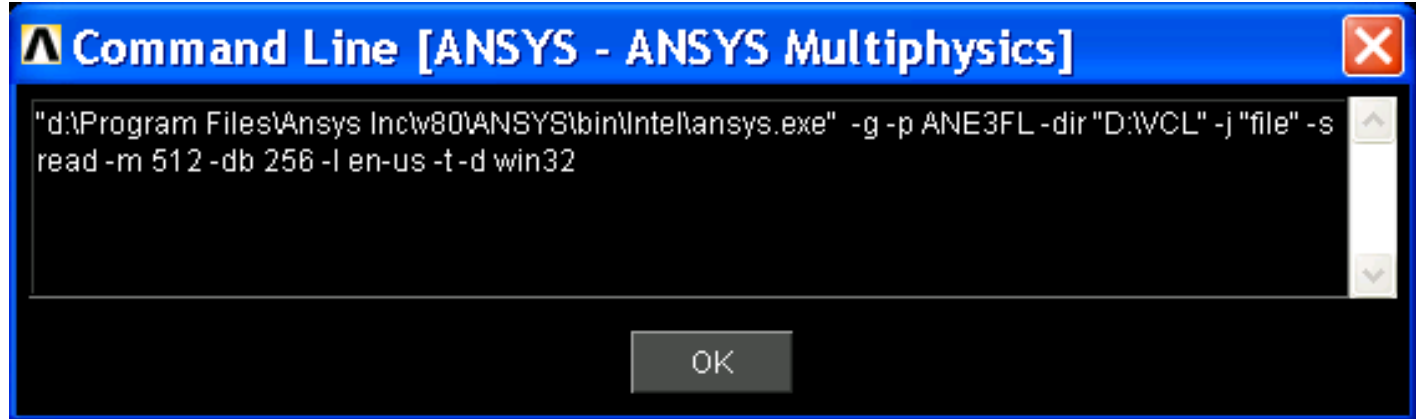
`SW_SHOWMAXIMIZED` – развернутое на весь экран.

Помимо WinExec, может также использоваться функция CreateProcess (см. далее).

Наиболее эффективный способ формирования командной строки для запуска ANSYS предполагает использование (однократное) ANSYS Product Launcher. После определения необходимых параметров запуска ANSYS (таких как рабочая директория, имя файла, размер базы данных и т.п.), необходимо

выбрать пункт меню Tools → Display Command Line.

В диалоговом окне будет приведен текст командной строки.



Необходимо отметить, что приложение `ansys.exe` является консольным.

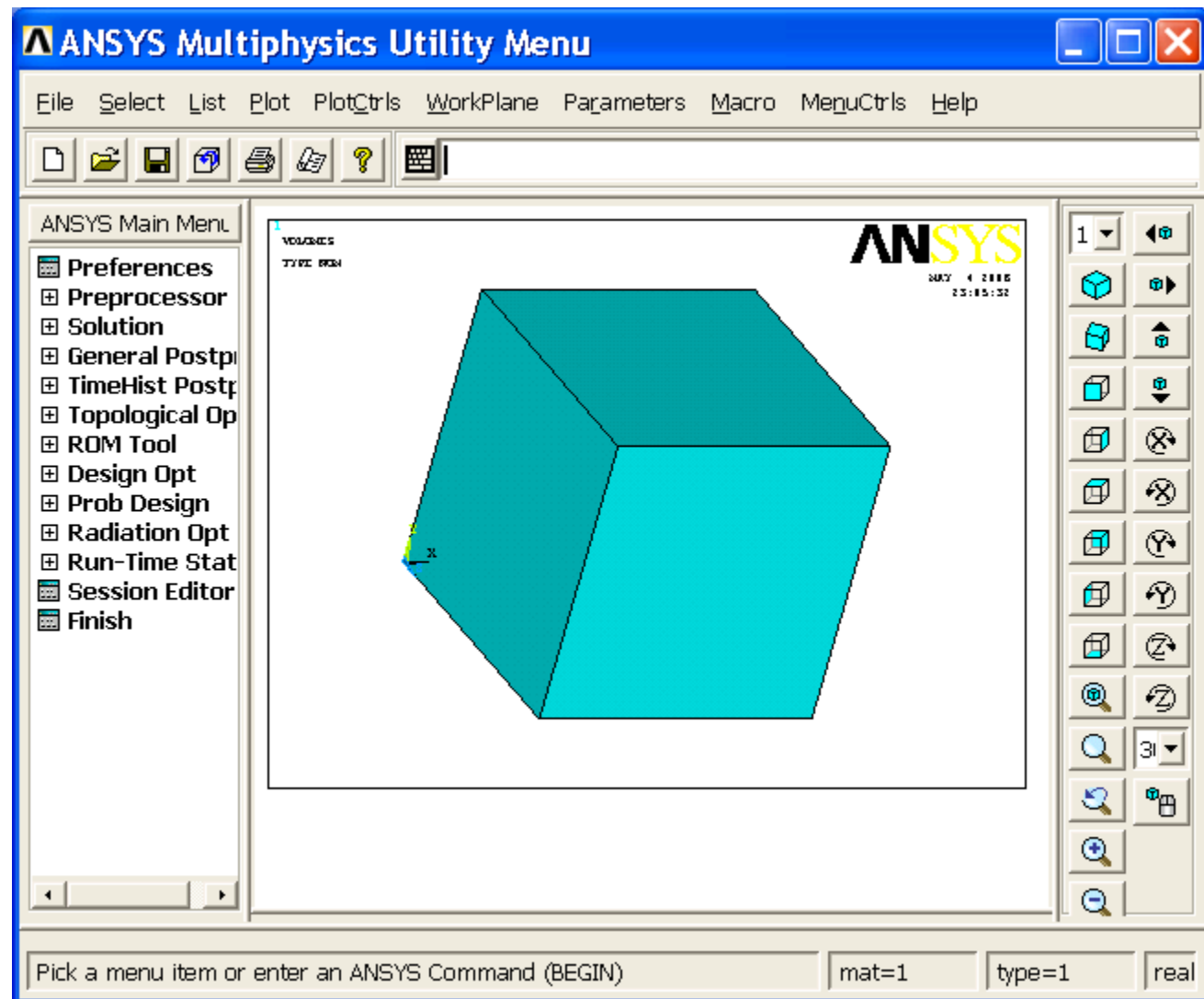
После запуска, `ansys.exe` сначала выполнит команды содержащиеся в файле `startXX.ans`, и только затем запустит графическую оболочку.

Следствием этого является невозможность выполнения команд, содержащихся в `startXX.ans`, касающихся вывода на экран в интерактивном режиме. Простейшим примером интеграции с ANSYS в интерактивный режим может служить

СЛЕДУЮЩИЙ ФРАГМЕНТ ПРИЛОЖЕНИЯ:

```
procedure TForm1.Button1Click(Sender:
    TObject);
var f:textfile;
begin
    assignfile(f,'start80.ans');
    rewrite(f);
    writeln(f,'/PREP7');
    writeln(f,'BLC4,0,0,1,1,1');
    writeln(f,'/RGB,INDEX,100,100,100,0');
    writeln(f,'/RGB,INDEX,0,0,0,15');
    writeln(f,'/ANG, 1 ,30.000000,XS,1');
    writeln(f,'/ANG, 1 ,30.000000,YS,1');
    closefile(f);
    winexec(PChar('"d:\Program Files\Ansys
        Inc\v80\ANSYS\bin\Intel\ansys.exe" -g
        -p ANE3FL -dir "D:\AnsProjects\iks" -j
        "file" -s read -d win32'),1);
end;
```

Вид ANSYS после ввода команды vplot.



При работе с ANSYS в пакетном режиме, может быть использована передача данных как от пользовательского приложения к ANSYS, так и в обратном направлении.

Принципы передачи данных от пользовательского приложения к ANSYS не отличаются от рассмотренных выше для работы с ANSYS в интерактивном режиме. Входной файл, при этом, в отличие от интерактивного режима, может иметь любое имя.

Алгоритм передачи данных в обратном направлении выглядит следующим образом:

- 1. Ожидание завершения работы ANSYS.**
- 2. Чтение файлов, созданных или модифицированных в результате работы ANSYS.**
- 3. Обработка (или вывод) информации.**

При этом предполагается, что во входном файле содержатся команды ANSYS для создания либо изменения файлов с результатами работы.

Командная строка для запуска ANSYS в пакетном режиме имеет следующую структуру:

`<путь к исполняемым файлам ANSYS>\
ansysXX.exe -b -i <имя входного
файла> -o <имя выходного файла>`

Например:

`"d:\Program Files\Ansys Inc\ ↵
v80\ANSYS\bin\Intel\ansys80.exe" ↵
-b -i 1.in -o 1.out`

Для ожидания завершения работы ANSYS, как и любого другого процесса в операционной системе необходимо знать описатель (handle) процесса.

Функция Windows API

- **function WaitForSingleObject(hHandle: THandle; dwMilliseconds: DWORD): DWORD; stdcall;**
приостанавливает работу приложения ее вызвавшего до тех пор пока не будет освобожден объект ядра Windows (для

**процесса - пока он не будет завершен),
описатель которого передан в качестве
первого параметра.**

**Второй параметр определяет интервал
времени в миллисекундах, по истечении
которого приложение продолжит работу,
даже если ожидаемый объект ядра не
будет освобожден. Если в качестве
параметра `dwMilliseconds` передать
константу `INFINITE`, то приложение не
продолжит работу до освобождения
объекта.**

Еще одна функция Windows API

- **function CreateProcess(lpApplicationName: PChar; lpCommandLine: PChar; lpProcessAttributes, lpThreadAttributes: PSecurityAttributes; bInheritHandles: BOOL; dwCreationFlags: DWORD; lpEnvironment: Pointer; lpCurrentDirectory: PChar; const lpStartupInfo: TStartupInfo; var lpProcessInformation: TProcessInformation): BOOL; stdcall;**

позволяет запустить процесс и получить его описатель.

Большинство параметров этой функции, как правило, не используются (т.е. передаются значения 0 или nil).

Параметр IpCommandLine определяет командную строку для запуска процесса. Параметр IpStartupInfo определяет параметры запуска. Тип TStartupInfo представляет собой тип-запись. Поле cb этой структуры несет информацию о ее размере. Остальные поля для запуска ANSYS могут иметь нулевые значения. Но, в

любом случае, все поля IpStartupInfo должны быть обязательно определены. После выполнения функции, в случае успешного запуска процесса, результат функции будет иметь значение true, а параметр IpProcessInformation содержать информацию об описателях и идентификаторах процесса и его основного потока.

Когда описатели объектов становятся

ненужными, они должны быть закрыты функцией

- **function CloseHandle(hObject: THandle): BOOL; stdcall;**

Параметр hObject определяет описатель, который должен быть закрыт.

Необходимо заметить, что после запуска CreateProcess нужно закрыть не только дескриптор процесса, но и его основного потока.

Следующий фрагмент кода иллюстрирует основные принципы интеграции с ANSYS в пакетном режиме:

```
procedure TForm1.Button1Click(Sender: TObject);
var si:TStartupInfo;
    pi:TProcessInformation;
    f:textfile;
begin
    assignfile(f, '1.in');
    rewrite(f);
    writeln(f, '/PREP7');
    writeln(f, 'BLC4,0,0,1,1,1');
    writeln(f, '/SHOW,JPEG');
    writeln(f, '/RGB,INDEX,100,100,100,0');
    writeln(f, '/RGB,INDEX,0,0,0,15');
    writeln(f, '/ANG, 1 ,30.000000,XS,1');
    writeln(f, '/ANG, 1 ,30.000000,YS,1');
```

```
writeln(f, 'VPLO') ;  
writeln(f, '/SHOW,CLOSE') ;  
closefile(f) ;  
ZeroMemory(@si, sizeof(TStartupInfo)) ;  
si.cb:=sizeof(TStartupInfo) ;  
if CreateProcess(nil, PChar('"d:\Program  
Files\Ansys Inc\v80\ANSYS\bin\Intel  
\ansys80.exe" -b -i 1.in -o  
1.out'), nil, nil, true, 0, nil, nil, si, pi)  
then  
begin  
    WaitForSingleObject(pi.hProcess, INFINITE) ;  
    image1.Picture.LoadFromFile('file000.jpg') ;  
    CloseHandle(pi.hThread) ;  
    CloseHandle(pi.hProcess) ;  
end ;  
end ;
```

Вид приложения после нажатия на кнопку

