

лее целостного и гармоничного специалиста, способного не только эффективно действовать на своем рабочем месте, но и хорошо решающего свои личностные проблемы, понимающего и умеющего грамотно и правильно организовывать свой отдых и работу.

ЛИТЕРАТУРА

1. Основы современной философии. - Санкт-Петербург, 1997.
2. Уроки китайской гимнастики. Вып. 3. А.Л. Маслов, А.М. Подщеколдин. - Москва: Советский спорт, 1990.
3. Искусство ци-гун Вон Кьют-Кит. - Москва, 1999.
4. Тайдзи-цигун янзюньшин. - Москва, 1999.
5. Цигун и спорт. - 1991.- № 2.

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК ДИНАМИЧЕСКОЙ КОМПОНОВКИ ПРИ ПРОГРАММИРОВАНИИ НА РАЗНЫХ ЯЗЫКАХ

Р.Б. Голубцов

Гомельский государственный технический университет им. П. О. Сухого

Б. И. Голубцов, к.ф.-м.н.

Гомельский филиал МНИТСО

Использование программ, написанных на различных языках, – очень распространенная технология программирования при создании серьезных программных комплексов. Зачастую бывает очень полезно использовать возможности того языка программирования (ЯП), на котором наиболее удобно можно описать конкретный алгоритм, если основным языком при создании пакета программ является другой. Например, если за основной язык принят Си, но требуется использование подпрограммы, оперирующей данными комплексного типа, то эту подпрограмму целесообразно написать на языке Фортран-77 (Фортран-90), т. к. в этом ЯП наиболее удобно реализована поддержка данных с таким типом. Другой пример: основной язык – Фортран-77 (Фортран-90); алгоритм головной программы требует использования подпрограмм с переменным числом параметров. Идеальный вариант – оформить такую подпрограмму в виде Си-функции с переменным числом параметров.

Проблемам сопряжения программ, написанных на разных ЯП, функционирующих под управлением операционной системы DOS (MS-DOS, PC-DOS, DR-DOS, Novell DOS и др.), авторами посвящена работа [2]. При создании DOS-программ сопряжение разноязыковых модулей осуществляется на этапе генерации компоновщиком исполнимого модуля программы. Несколько иной подход к технологии построения разноязыковых программ может быть использован, когда разрабатывается программная система, предназначенная для работы под управлением операционной системы Microsoft® Windows®, в частности, любой операционной системы из семейства Microsoft® Win32® (Microsoft® Windows® 95, Windows® 98, Windows® 2000 и Windows NT®). В этом случае очень удобно использовать библиотеки динамической компоновки. Главное отличие здесь от предыдущего подхода заключается в том, что сопряжение происходит во время выполнения программы, а не во время ее построения (генерации). Однако при создании Windows-программ можно использовать и старый механизм сопряжения.

При создании программ, в которых применяется механизм динамической компоновки, компоновщик генерирует исполнимый модуль (основную программу) и одну или несколько динамически подключаемых библиотек с программами как основного, так и других ЯП. Как обычно, имя исполнимого модуля имеет расширение .EXE, а имя модуля динамической компоновки – расширение .DLL. Во время своей работы исполнимый модуль функционирует, выполняя заложенный в него алгоритм до тех пор, пока не будет вызвана подпрограмма, содержащаяся в библиотеке динамической компоновки. При вызове такой подпрограммы в память компьютера загружается библиотека и выполняется нужная подпрограмма, после чего библиотека выгружается из памяти, а основная программа продолжает свою работу по-прежнему. Такой процесс называется *динамической компоновкой*. Он повторяется всякий раз, когда вызывается подпро-

грамма из модуля динамической компоновки. Механизм динамической компоновки достаточно полно рассмотрен в книге [1].

Сказанное можно проиллюстрировать примерами. В данном случае авторы использовали следующие алгоритмические языки программирования и соответствующие им компиляторы:

язык Си (компилятор Microsoft® 32-bit C/C++ Optimizing Compiler V11.0 for 80x86 из системы программирования Microsoft® Visual C++® 5.0);

язык Фортран-90 (компилятор Microsoft® 32-bit Fortran Optimizing Compiler V4.0 из системы программирования Microsoft® Fortran PowerStation 4.0).

В первом примере программа, написанная на языке Си, вызывает функцию (аналогично можно вызвать и процедуру), написанную на языке Фортран-90, из библиотеки динамической компоновки. Второй пример демонстрирует обратный вызов: программа, написанная на языке Фортран-90, вызывает функцию, написанную на языке Си, которая содержится в библиотеке динамической компоновки. Исходные тексты исполнимого и библиотечного программных модулей, а также тексты файлов описания проектов для каждого примера приводятся ниже.

Пример 1. Вызов Фортран-90-подпрограммы из Си-программы

Программа на языке Си (test_c.c)

```
#include <windows.h>
typedef INT (*F_F90)(INT, INT);
VOID main(VOID)
{
    static INT i_value1, i_value2, i_value3;
    HINSTANCE hDLL;
    F_F90 Function_F90;
    printf("\nЗначение 1: ");
    scanf("%i", &i_value1);
    printf("Значение 2: ");
    scanf("%i", &i_value2);
    hDLL = LoadLibrary("func_f90.dll");
    Function_F90 = (F_F90)GetProcAddress(hDLL, "Func");
    i_value3 = (Function_F90)(i_value1, i_value2);
    FreeLibrary(hDLL);
    printf("Значение 3 = %i\n", i_value3);
}
```

Подпрограмма на языке Фортран-90 (func_f90.f90)

```
integer function Func(a, b)
!ms$ attributes stdcall, alias: "Func@16" :: Func
implicit none
integer a, b
    Func = a * b
    return
end function Func
```

Файл определения модуля (func_f90.def)

```
EXPORTS
    Func
```

Файл описания проекта (makefile)

```
PROJECT_DLL = func_f90
PROJECT_EXE = test_c
FC = fl32
```

```

CC = cl
LINK = link
FFLAGS = -Ic:\users\boris\msfps\include -c
CFLAGS = -Ic:\users\boris\msc\include -c
LFLAGS_DLL = -LIBPATH:c:\users\boris\msfps\lib -
DEF:$(PROJECT_DLL).def -DLL -RELEASE -VERSION:3.51
LFLAGS_EXE = -LIBPATH:c:\users\boris\msc\lib -RELEASE -
VERSION:3.51
OBJS_DLL = $(PROJECT_DLL).obj
OBJS_EXE = $(PROJECT_EXE).obj
all: $(PROJECT_DLL).dll
all: $(PROJECT_EXE).exe
.f90.obj:
    $(FC) $(FFLAGS) $*.f90
.c.obj:
    $(CC) $(CFLAGS) $*.c
$(PROJECT_DLL).dll: $(OBJS_DLL)
    $(LINK) $(LFLAGS_DLL) $(OBJS_DLL)
$(PROJECT_EXE).exe: $(OBJS_EXE)
    $(LINK) $(LFLAGS_EXE) $(OBJS_EXE)

```

Здесь Си-программа использует системные функции: LoadLibrary загружает динамически компонуюемую библиотеку в память, GetProcAddress извлекает физический адрес памяти Фортран-90-подпрограммы. Получив адрес, программа выполняет данную подпрограмму, после чего выполняется системная функция FreeLibrary, которая выгружает библиотеку из памяти. Такой механизм динамической компоновки получил название *компоновка во время выполнения программы*. Другой тип динамической компоновки иллюстрирует пример 2.

Пример 2. Вызов Си-подпрограммы из Фортран-90-программы
Программа на языке Фортран-90 (test_f90.f90)

```

program Test
implicit none
interface
    integer function Func(a, b)
    !ms$ attributes stdcall, alias: "_Func" :: Func
    integer a, b
    end function
end interface
integer i_value1, i_value2, i_value3
write(*, "(/A13\)" ) "Значение 1: "
read(*, *) i_value1
write(*, "(A13\)" ) "Значение 2: "
read(*, *) i_value2
i_value3 = Func(i_value1, i_value2)
write(*, "(A14I10)" ) "Значение 3 = ", i_value3
end

```

Подпрограмма на языке Си (func_c.c)

```

int Func(int a, int b)
{
    return a * b;
}

```

Файл определения модуля (func_c.def)

EXPORTS

Func

Файл описания проекта (makefile)

```
PROJECT_DLL = func_c
PROJECT_EXE = test_f90
CC = cl
FC = fl32
LINK = link
CFLAGS = -Ic:\users\boris\msc\include -c
FFLAGS = -Ic:\users\boris\msfps\include -c
LFLAGS_DLL = -LIBPATH:c:\users\boris\msc\lib -
DEF:$(PROJECT_DLL).def -DLL -RELEASE -VERSION:3.51
LFLAGS_EXE = -LIBPATH:c:\users\boris\msfps\lib -RELEASE -
VERSION:3.51
OBJS_DLL = $(PROJECT_DLL).obj
OBJS_EXE = $(PROJECT_EXE).obj
LIBS = $(PROJECT_DLL).lib
All: $(PROJECT_DLL).dll
All: $(PROJECT_EXE).exe
.c.obj:
    $(CC) $(CFLAGS) *.c
.f90.obj:
    $(FC) $(FFLAGS) *.f90
$(PROJECT_DLL).dll: $(OBJS_DLL)
    $(LINK) $(LFLAGS_DLL) $(OBJS_DLL)
$(PROJECT_EXE).exe: $(OBJS_EXE)
    $(LINK) $(LFLAGS_EXE) $(OBJS_EXE) $(LIBS)
```

В данном примере применен механизм динамической компоновки, называемый *компоновкой во время загрузки программы в память*. Программа, написанная на языке Фортран-90, загружаясь в память, загружает с собой и динамически подключаемую библиотеку, содержащую Си-подпрограмму. При завершении работы, головная программа выгружается из памяти, а с ней выгружается и библиотека.

Использование библиотек динамической компоновки при программировании на различных языках дает заметный выигрыш в скорости во время работы программы по сравнению со статической компоновкой. Но главное достоинство такого подхода заключается в том, что программист, создав такую библиотеку, может распространять ее как готовую программу, будучи уверенным, что обращение к ней из программ, написанных на других ЯП и другими программистами, будет осуществляться безошибочно.

ЛИТЕРАТУРА

1. Гладков С. А., Фролов Г. В. Программирование в Microsoft Windows: В 2-х ч. Ч. 1, 2. – М.: ДИАЛОГ-МИФИ, 1992. – 320 с., 288 с.
2. Голубцов Б. И., Голубцов Р. Б. О сопряжении разноязыковых программных модулей // Концепции социально-экономического развития Беларуси: черты, закономерности, перспективы: Материалы науч.-практ. конф. (19 мая 1999 г., г. Гомель) / Под ред. С. И. Ляха, Н. С. Щупловой. – Гомель: МНИТСО, Гомельский филиал, 2000. – С. 115 – 120.