

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ

Учебный курс

МЕТОДЫ ПРОГРАММИРОВАНИЯ - 2





Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Учебный курс:

Методы программирования - 2

Практическая работа 5:

*Структуры хранения нескольких стеков в
общей памяти*

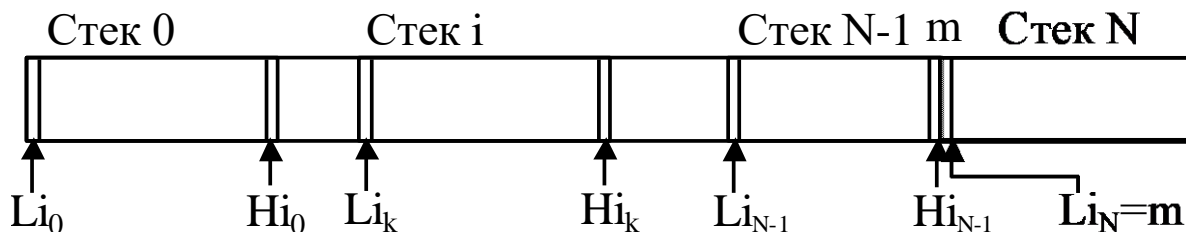
Гергель В.П., профессор ,
директор института ИТММ

Содержание

Разработка системы поддержки нескольких стеков

- Структура памяти и ее свойства
- Начальное распределение памяти
- Оценка имеющейся свободной памяти
- Перераспределение памяти на основе гипотез о поведении стеков
- Реализация: Вставка и выборка значений
- Реализация: Перепаковка структуры хранения

1. Структура памяти и её свойства



где N – количество стеков, m – размер памяти

Свойства

- ✓ $Li_0=0$ - условие неподвижности 1 стека
- ✓ $Hi_k=Li_k-1$ - условие пустоты
- ✓ $Hi_k < Li_{k+1}$ - условие неперекрывания
- ✓ $Hi_k=Li_{k+1}-1$ - условие переполнения

Для выполнимости последних двух условий для всех стеков введем фиктивный стек N, для которого $Li_N=m$.

2. Начальное распределение памяти

Будем предполагать, что все стеки используются с одинаковой интенсивностью \Rightarrow память распределяется всем стекам поровну

- $Li_0=0, Li_N=m$
- $Li_k=Li_{k-1}+ m / N, 1 \leq k < N$

3. Перепаковка – оценка свободной памяти

Выполняется при попытке вставки нового значения в стек s , у которого отсутствует свободная память

$$F = \sum (Li_k - Hi_{k-1} - 1), \quad 1 \leq k < N$$

☑ $F = 0$ – свободной памяти нет

☑ $F = 1$ – свободен 1 элемент памяти и его следует отдать стеку s

☑ $F > 1$ – необходимо перераспределить свободную память

Для гарантированного выделения свободной памяти стеку s при наличии только одного свободного элемента памяти (случай 2), выполним

$-Hi_s = Hi_s + 1$ – перед началом процедуры оценки свободной памяти

$-Hi_s = Hi_s - 1$ – после завершения перепаковки



4. Перепаковка – распределение свободной памяти

Снова предположим, что все стеки используются с одинаковой интенсивностью \Rightarrow свободная память должна распределиться всем стекам поровну

$$- Li'_0 = Li_0, Li'_N = Li_N$$

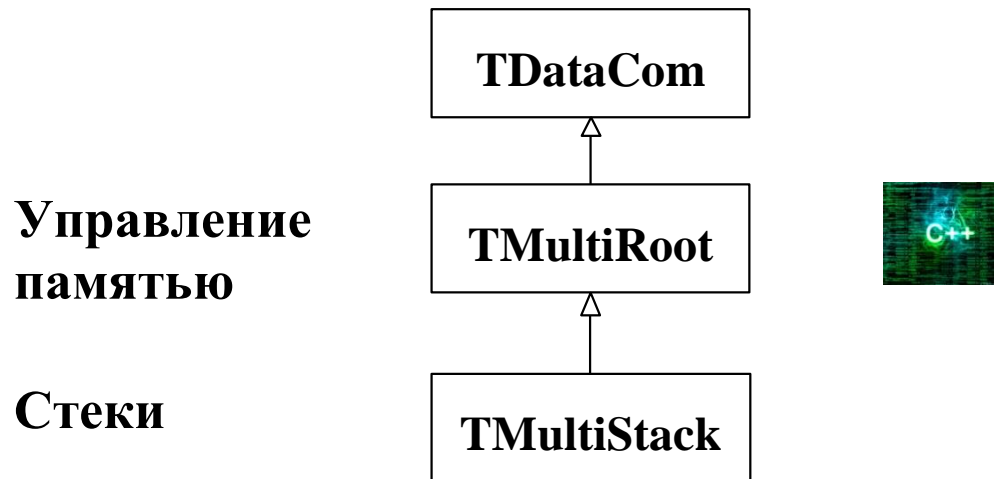
$$- Li'_k = Li'_{k-1} + (Hi_{k-1} - Li_{k-1} + 1) + F / N, 1 \leq k < N$$

☑ На равномерность распределения памяти может сказаться целочисленность операции деления

5.1. Реализация – схема наследования

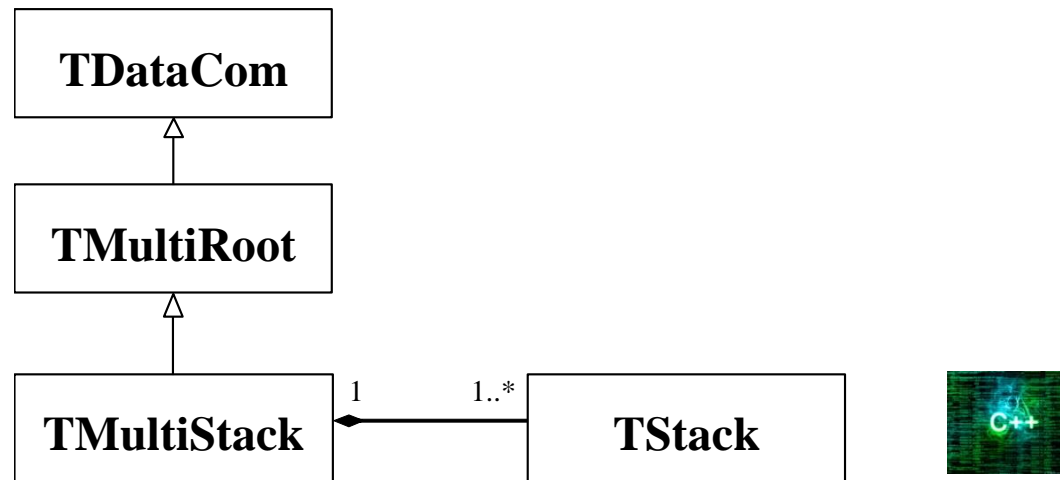
Реализацию системы можно снова разделить на 2 этапа:

- разработка абстрактного базового класса **TMultiRoot** для управления памятью и определения спецификаций необходимых операций
- разработка класса **TMultiStack** для обеспечения работы с несколькими стеками



5.2. Реализация – вставка и выборка значений

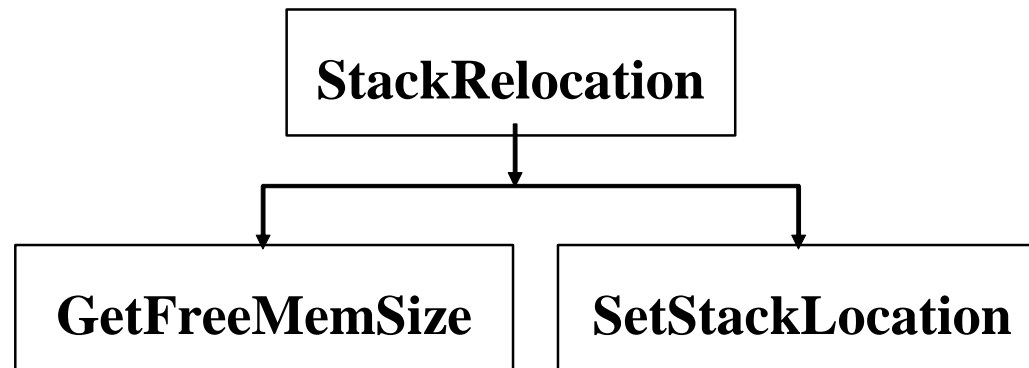
Для работы с отдельным стеком используется ранее разработанный класс **TStack** (!) – такая возможность обеспечивается ранее предусмотренной возможностью передачи извне памяти для хранения значений (метод **SetMem**)



5.3. Реализация – перепакровка ...

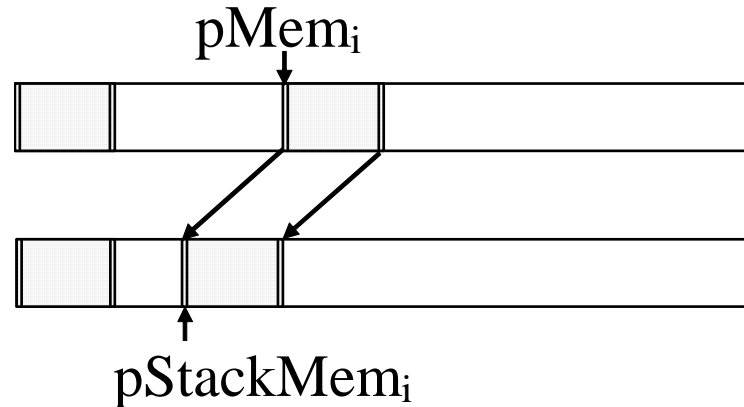
В соответствии с последовательностью действий, необходимых для выполнения перепакровки, определим следующий набор методов для динамического перераспределения памяти:

- **StackRelocation** – основной метод перепакровки;
- **GetFreeMemSize** – оценка свободной памяти;
- **SetStackLocation** – планирование нового расположения стеков (после перепакровки)
- **GetRelocationCount** – получение количества перепакровок за все время работы системы

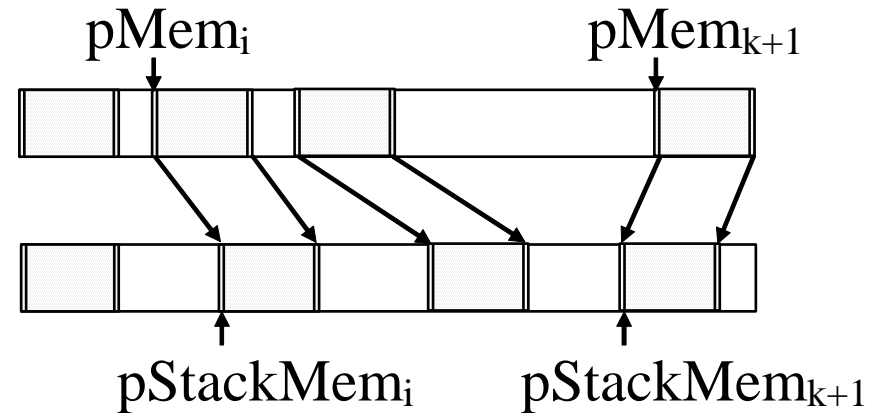


5.4. Реализация – перепаковка

Перемещение стека влево



Перемещение стека вправо



Контрольный пример:



Заключение

- Структура памяти для размещения нескольких стеков
- Начальное распределение памяти
- Ситуация локального переполнения памяти
- Оценка свободной памяти
- Динамическое перераспределение памяти при помощи перепаковки данных
- Схема наследования и последовательность разработки программ
- Реализация перепаковки

Вопросы для обсуждения

- Стратегии динамического перераспределения памяти
- Модульность действий, выполняемых при перепакровке (оценка свободной памяти, стратегия перераспределения памяти,...)
- Использование ранее разработанного ПО (класс TStack)

Темы для самостоятельной работы

- Разработка тестов для проверки правильности работы программ
- Динамическое изменение количества стеков

Следующая тема

- Гипотезы поведения стеков для определения стратегии динамического перераспределения памяти

Контакты

Нижегородский государственный университет им.
Н.И. Лобачевского (www.unn.ru)

Институт информационных технологий, математики
и механики (www.itmm.unn.ru)

603950, Нижний Новгород, пр. Гагарина, 23,
р.т.: (831) 462-33-56,

Гергель Виктор Павлович

(<http://www.software.unn.ru/?dir=17>)

E-mail: gergel@unn.ru

