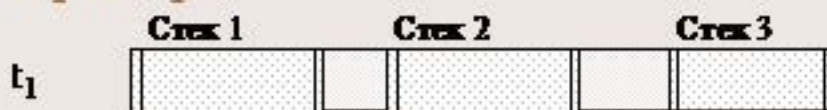


1. Динамическое распределение памяти...

Определение 1.13. Распределение памяти до начала процесса вычислений называется *статическим*. Распределение памяти в ходе выполнения программы называется *динамическим распределением памяти*.

Пример.



В общей области памяти располагаются 3 стека



Стек 1 переполнен



Перераспределение свободной памяти путем перемещения стеков по памяти

Определение 1.14. Процедура динамического перераспределения памяти путем переписи части хранимых значений в другую область памяти называется *перепакеткой памяти* или просто *перепакеткой*.

1. Динамическое распределение памяти

- ☑ Перепаковка обеспечивает эффективное использование одного ресурса ЭВМ (*памяти*) за счет другого ресурса (*времени*).
- ☑ Для выполнения перепаковки требуется разработка управляющих программ.

Определение 1.15. Выполнение функций анализа свободной памяти, планирование размещения структур, переписывание структур называется *управление памятью*. Комплекс программ, реализующих управление памятью называется *системой управления памятью*.

- ☑ Необходимость перепаковки обуславливается принятым способом реализации отношений следования.

3. Роль гипотез о поведении стеков...

- ☑ Гипотезы о поведении структур служат основой для принятия решений о распределении памяти
- ☑ Формирование гипотез происходит в результате теоретического анализа модели решаемой задачи или может быть выполнено на основе статистических данных, получаемых в ходе вычислительных экспериментов с проектируемой программной системой
- ☞ **Гипотеза 1:** Стеки используются с одинаковой интенсивностью
⇒ память разделяется между стеками поровну
- ☞ **Гипотеза 2:** Интенсивность использования стеков различается. Конструктивное предположение о характере такой неравномерности может состоять в гипотезе *сохранения локальных тенденций роста* стеков, т.е. в каждый момент времени использование стеков на последующих шагах вычислений характеризуется точно таким же поведением, что и на предшествующих этапах обработки данных.

3. Роль гипотез о поведении стеков...

☞ Гипотеза 2: Сохранение локальных тенденций роста

- показатель роста стека

$$\delta_i = \max (0, \text{DataCount}'_i - \text{DataCount}_i)$$

- суммарный показатель роста

$$\Delta = \sum \delta_i, 0 \leq i < N$$

- правило распределение памяти для стеков в соответствии с их показателями роста

$$\text{Li}'_k = \text{Li}'_{k-1} + (\text{Hi}_{k-1} - \text{Li}_{k-1} + 1) + F^*(\delta_i / \Delta), 1 \leq k < N$$

⌚ Как изменить программы системы для применения нового варианта перераспределения памяти ?

☞ Достаточно переопределить метод планирования памяти для **перепакетки SetStackLocation** (процедуру оценки показателей роста целесообразно выделить в отдельный метод **SetStackRate**)

3. Роль гипотез о поведении стеков...

↪ **Гипотеза 3:** Использование вероятностных предположений о поведении стеков

Пусть есть θ , $0 \leq \theta \leq 1$, вероятность выполнимости гипотезы сохранения локальных тенденций роста. Тогда

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

Распределение памяти
поровну между стеками

Распределение памяти
пропорционально
показателям роста

⌚ Какая дополнительная доработка системы требуется для применения новой схемы распределения памяти ?

↪ Достаточно переопределить метод планирования памяти для перепакетки `SetStackLocation` (!)

3. Роль гипотез о поведении стеков...

✎ Адаптивная оценка параметров модели...

⌚ Как оценить вероятность выполнимости гипотезы сохранения локальных тенденций роста θ ?

☑ Пусть σ есть количество выполненных перепакетов памяти за некоторый отрезок времени Δt . Величина σ зависит от значения θ и для повышения эффективности функционирования системы следует определить такое θ , чтобы количество перепакетов было минимальным, т.е.

$$\min_{\theta} \sigma(\theta)$$

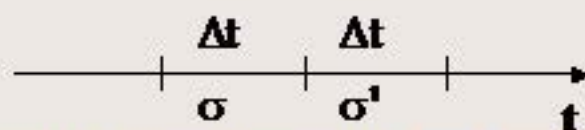
3. Роль гипотез о поведении стеков...

↳ Адаптивная оценка параметров модели...

$$\min \sigma(\theta)$$

☑ Возможная схема определения оптимального значения θ может состоять в следующем:

- выполним оценку величины σ на последовательных друг за другом отрезках времени Δt



и определим величину изменения количества выполненных перепакетов:

$$\Delta \sigma = \sigma' - \sigma$$

- примем следующее следующее правило корректировки значения θ

$$\theta' = \begin{cases} \theta + \Delta\theta, & \Delta\sigma \leq 0, \\ \theta - \Delta\theta, & \Delta\sigma > 0, \end{cases}$$

где $\Delta\theta$ есть параметр схемы адаптации

3. Роль гипотез о поведении стеков...

↪ **Адаптивная оценка параметров модели...**

- ☑ Для включения в разработанную программную системы схемы адаптации по прежнему достаточно переопределить метод планирования памяти перепакровки **SetStackLocation (!)**

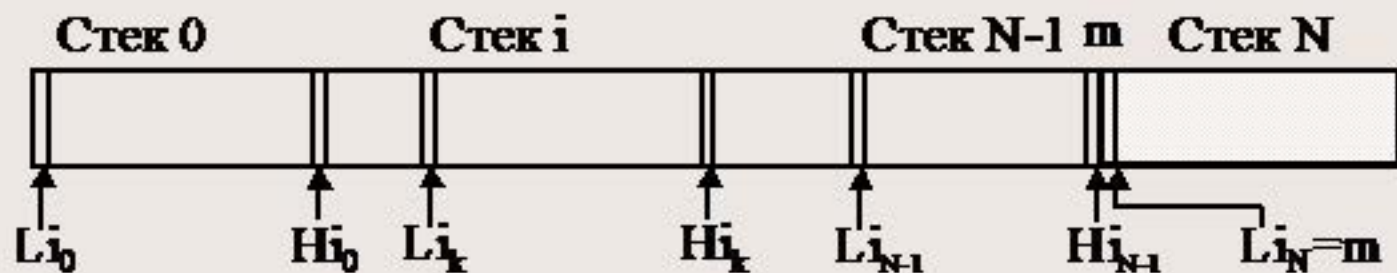


Контрольный пример: программа, приложение

Результаты вычислительных экспериментов

Гипотеза	Итераций	Количество перепакровок	Среднее	
Одинаковое использование	2007	1018	1, 97	
Сохранение тенденций роста	1195	726	1, 64	
Смешанная	2578	1337	1, 92	$\theta=0.5$
Адаптация	3031	1474	2, 06	$\theta=0.3$

1. Структура памяти и ее 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 неперекрывтия
- ☑ $Hi_k = Li_{k+1}-1$ - условие переполнения

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

3. Перепакровка - оценка свободной памяти

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

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

- ☑ $F = 0$ – 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 перепакровки

5.3. Реализация – перепаковка...

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

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

