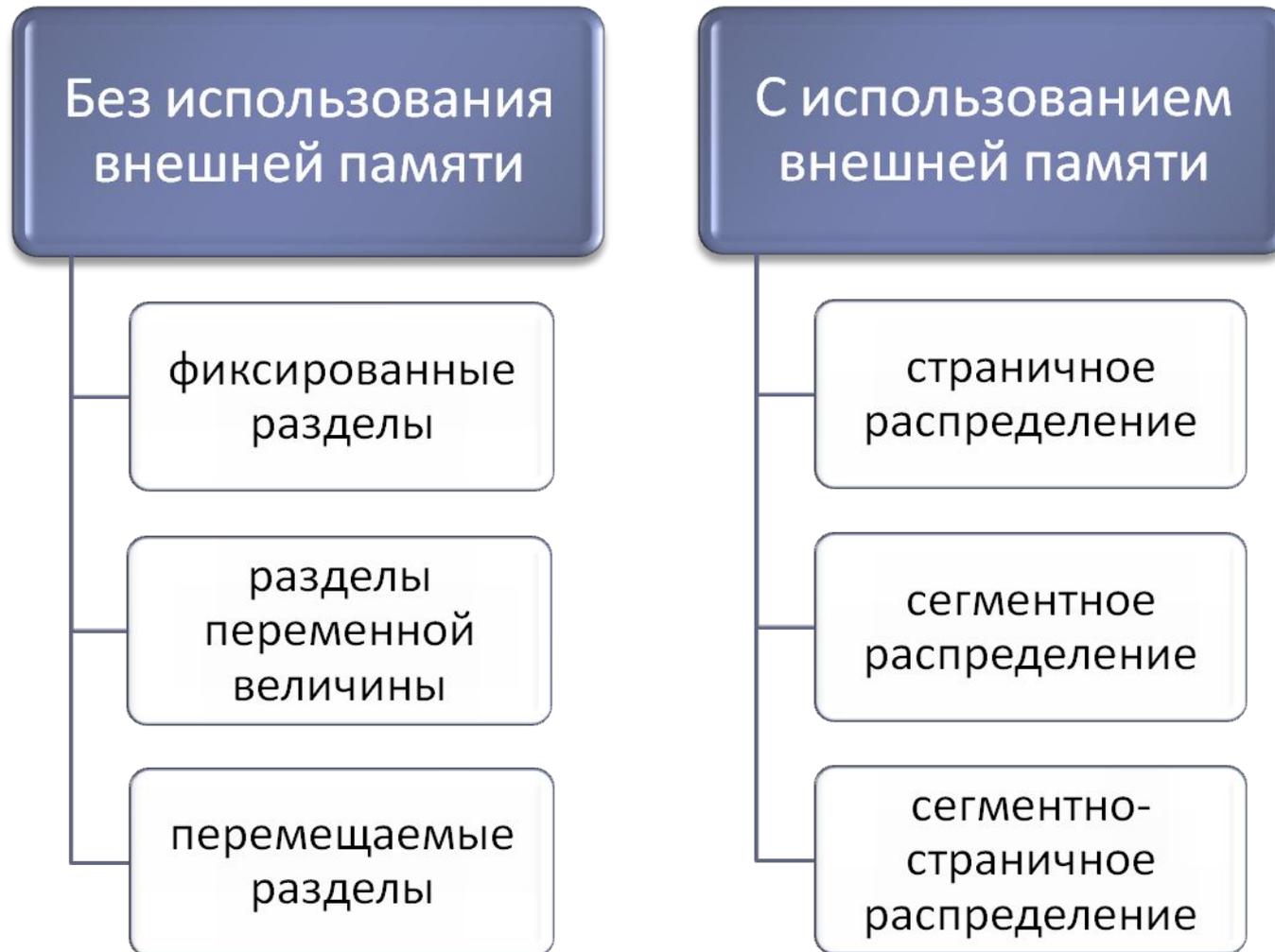


Операционные системы

Основы управления памятью

Классификация методов распределения памяти



Основы управления памятью

Методы распределения памяти без использования дискового пространства

Фиксированные разделы

- Самым простым способом управления оперативной памятью является разделение ее на несколько разделов фиксированного, но не обязательно равного размера (*Fixed Partitioning*).
- Разбиение памяти на разделы выполняется вручную оператором во время запуска системы, таким образом максимальное число одновременно выполняемых программ ограничено числом разделов вне зависимости от того, какой размер имеют программы (*Multiprogramming with a Fixed number of Tasks*).
- Эта схема была реализована в IBM OS/360, DEC RSX-11 и ряде других систем.
- Задачами операционной системы при реализации данного метода управления памятью является:
 - сравнение размера программы, поступившей на выполнение, и размера свободных разделов, и выбор подходящего раздела;
 - загрузка программы и настройку её адресов.



Достоинства и недостатки распределения фиксированными разделами

□ Достоинства:

- Простота реализации.
- Быстрое переключение с выполнения одной программы на другую (переключение контекста).

□ Недостатки:

- Жесткость реализации – в каждом разделе может выполняться только одна программа, то уровень мультипрограммирования заранее ограничен числом разделов (*Multiprogramming with a Fixed number of Tasks*).
- Внутренняя фрагментация – потеря части памяти, выделенной процессу, но не используемой им. Фрагментация возникает потому, что процесс не полностью занимает выделенный ему раздел.
- Программа сохраняется в разделе строго непрерывно.
- В случае если программа не может быть помещена целиком в выделенный ей раздел, то программист будет вынужден реализовать механизм оверлеев.

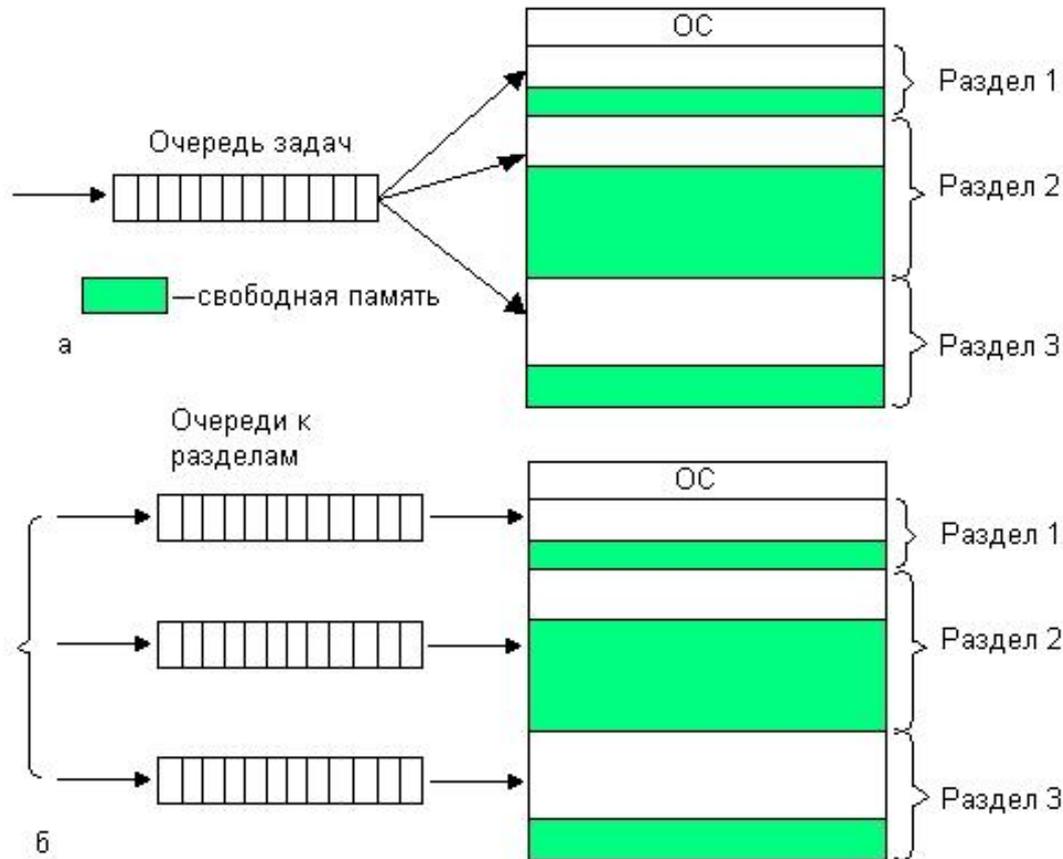


Алгоритмы распределение памяти фиксированными разделами (1)

- Разделы разного размера –
 - использование общей очереди
 - очередной процесс, поступивший на выполнение, помещается в общую очередь;
 - высокая внутренняя фрагментация;
 - использование нескольких очередей
 - для каждого размера раздела создается своя очередь;
 - стремится снизить внутреннюю фрагментацию;
 - однако очереди к некоторым разделам могут оказаться пустыми;
- Разделы равного размера –
 - общая очередь для всех процессов.



Алгоритмы распределение памяти фиксированными разделами (2)



а – общая очередь;

б – несколько отдельных очередей.



Разделы переменной величины (1)

- В случае использования разделов переменной величины оперативная память машины не делится заранее на разделы и исходно вся считается свободной (*Dynamic Partitioning*).
- Каждой вновь поступающей задаче выделяется необходимая ей память. Если достаточный объем памяти отсутствует, то задача не принимается на выполнение и стоит в очереди. После завершения задачи память освобождается, и на это место может быть загружена другая задача.
- Таким образом, в произвольный момент времени оперативная память представляет собой случайную последовательность занятых и свободных участков (разделов) произвольного размера.



Разделы переменной величины (2)

- Задачами операционной системы при реализации данного метода управления памятью является:
 - ведение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти;
 - при поступлении новой задачи – анализ запроса, просмотр таблицы свободных областей и выбор раздела, размер которого достаточен для размещения поступившей задачи (стратегии выбора раздела см. далее);
 - загрузка задачи в выделенный ей раздел (настройка адресов) и корректировка таблиц свободных и занятых областей;
 - после завершения задачи корректировка таблиц свободных и занятых областей.
- Программный код не перемещается во время выполнения, но может быть проведена единовременная настройка адресов посредством использования перемещающего загрузчика.



Достоинства и недостатки распределения разделами переменной величины

□ Достоинства:

- По сравнению с фиксированными разделами данный метод обладает гораздо большей гибкостью, количество одновременно выполняемых программ формально не ограничено.
- Разделы переменной величины используются процессами полностью, поэтому на уровне отдельных разделов полностью отсутствует фрагментация памяти (внутренняя фрагментация).

□ Недостатки:

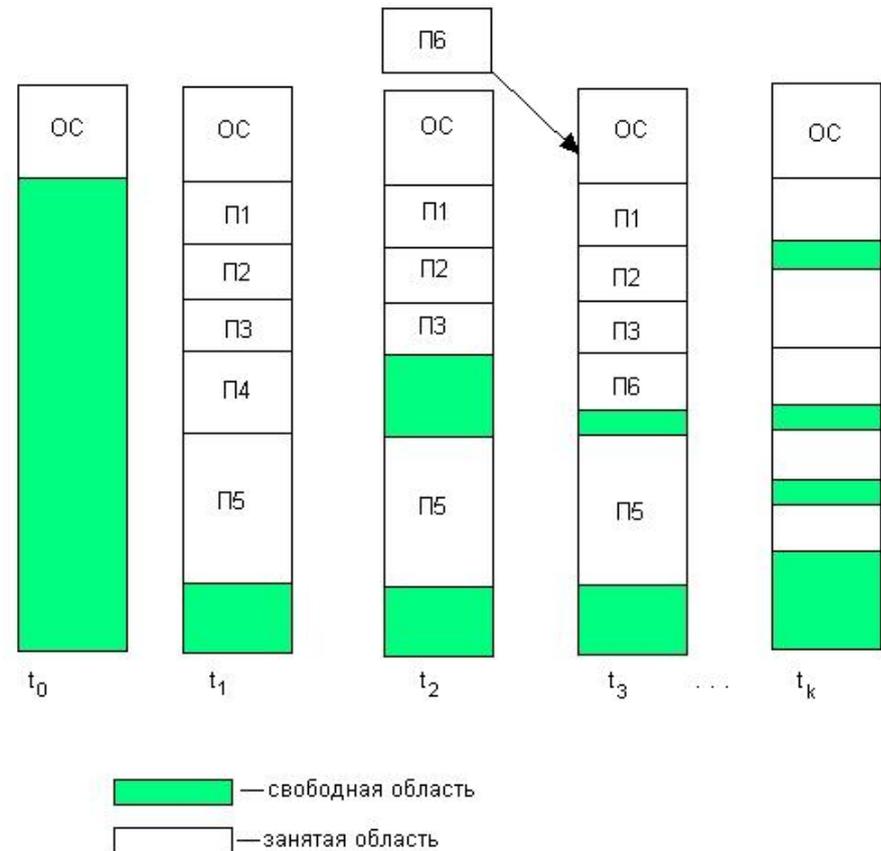
- Этому методу свойственна внешняя фрагментация памяти – наличие в оперативной памяти большого числа несмежных участков свободной памяти очень маленького размера (фрагментов). Настолько маленького, что ни одна из вновь поступающих программ не может поместиться ни в одном из участков, хотя суммарный объем фрагментов может составить значительную величину, намного превышающую требуемый объем памяти.

Иллюстрация внешней фрагментации



Иллюстрация внешней фрагментации

- На рисунке показано состояние памяти в различные моменты времени при использовании динамического распределения.
- В момент t_0 в памяти находится только ОС,.
- К моменту t_1 память разделена между 5-ю задачами, причем задача П4, завершаясь, покидает память.
- На освободившееся после задачи П4 место загружается задача П6, поступившая в момент t_3 .



Стратегии выбора свободного раздела

First fit

- стратегия первого подходящего

Best fit

- стратегия наиболее подходящего

Worst fit

- стратегия наименее подходящего

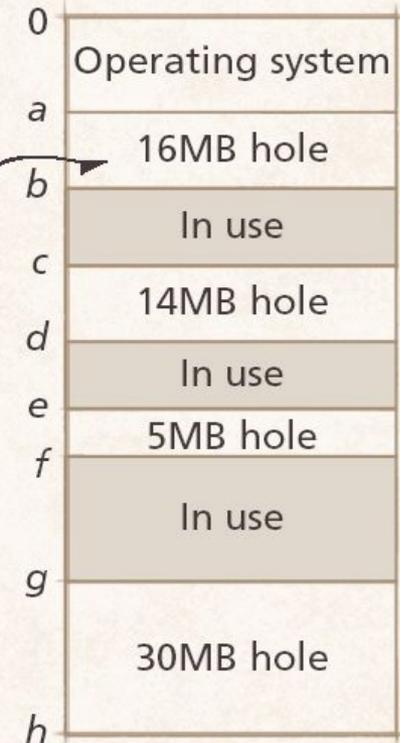
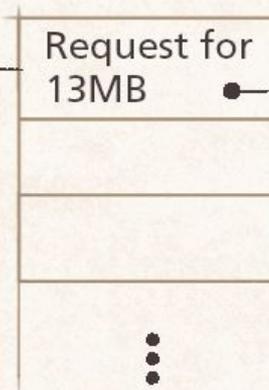


Стратегия первого подходящего

- Процесс помещается в первый подходящий по размеру раздел.

Free Memory List (Kept in random order.)

<i>Start address</i>	<i>Length</i>
a	16MB
e	5MB
c	14MB
g	30MB



Стратегия наиболее подходящего

- Процесс помещается в тот раздел, где после его загрузки останется меньше всего свободного места.

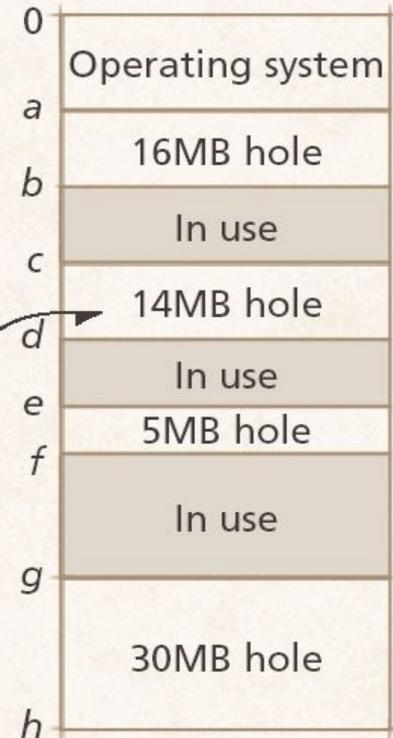
Free Memory List

(Kept in ascending order by hole size.)

Start address *Length*

e	5MB
c	14MB
a	16MB
g	30MB

Request for
13MB



Стратегия наименее подходящего

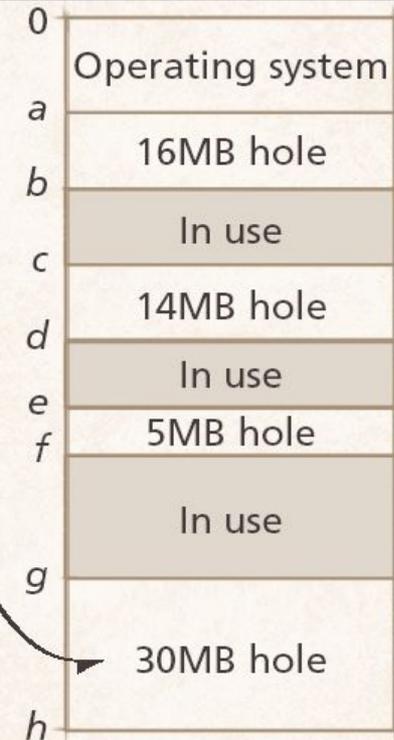
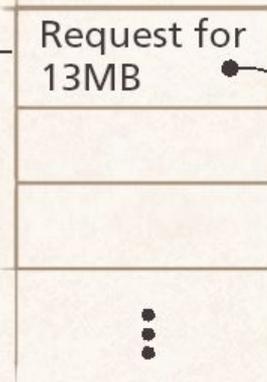
- Процесс помещается в самый большой раздел, в котором остается достаточно места для возможного размещения еще одного процесса.

Free Memory List

(Kept in descending order by hole size.)

Start address Length

<u>g</u>	30MB
a	16MB
c	14MB
e	5MB



Вопрос

- Какую из трех предложенных стратегий Вы считаете более эффективной?



Выбор эффективной стратегии

- Исследования показали, что доля полезно используемой памяти при использовании **First Fit** и **Best Fit** больше, при этом **First Fit** несколько быстрее.
- ☞ Перечисленные стратегии широко применяются и другими компонентами ОС, например для размещения файлов на диске.



Перемещаемые разделы (1)

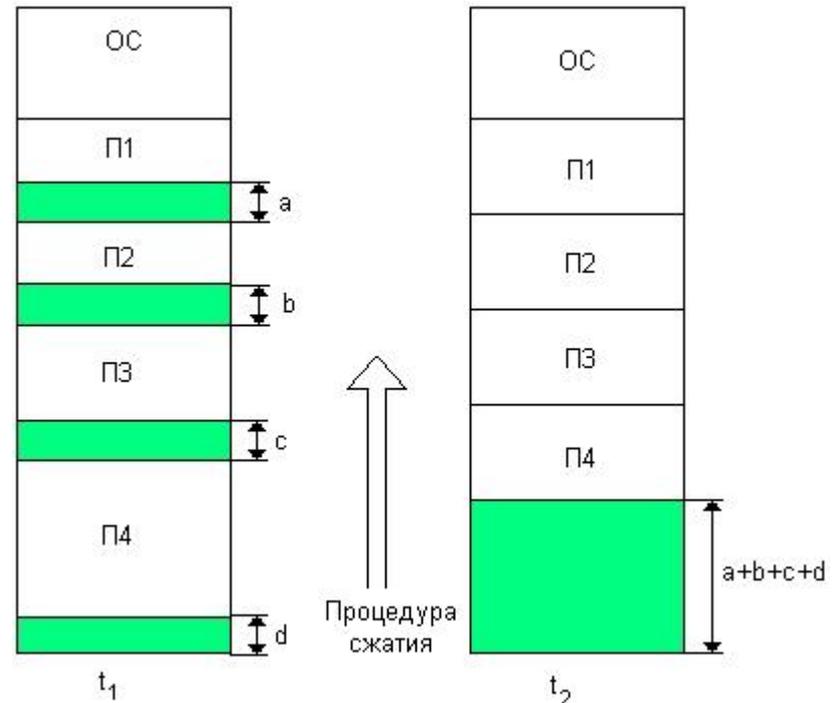
- Одним из методов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших либо в сторону младших адресов, так, чтобы вся свободная память образовывала единую свободную область (Relocatable dynamic partitions).
 - Эта процедура называется «сжатием» (compact или garbage collection), она требует от операционной системы время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей.
 - Так как программы перемещаются по оперативной памяти в ходе своего выполнения, то преобразование адресов должно выполняться динамическим способом.
-



Перемещаемые разделы (2)

□ Сжатие может выполняться либо при каждом завершении задачи, либо когда для новой задачи не хватает свободного раздела достаточного размера:

- в первом случае требуется меньше вычислительной работы при корректировке таблиц;
- во втором – реже выполняется процедура сжатия.



Достоинства и недостатки использования перемещаемых разделов

- Хотя процедура сжатия и приводит к более эффективному использованию памяти, она может потребовать значительного времени, что часто перевешивает преимущества данного метода.

