

# Управление реальной памятью

Операционные системы и среды,  
230105,230106

# СВОПИНГ

- **Свопингом (перекачкой) называется метод управления памятью, основанный на том, что все процессы, участвующие в мультипрограммной обработке, хранятся во внешней памяти.**
- **Процесс, которому выделен ЦП, временно перемещается в основную память (swap in/roll in).**
- **В случае прерывания работы процесса он перемещается обратно во внешнюю память (swap out/roll out). Причем целиком, а не отдельной частью.**

- Основное применение свопинг находит в системах разделения времени, где он используется одновременно со стратегией RR.
- В начале каждого временного кванта блок управления памяти выгружает из основной памяти процесс, работа которого была только что прервана, и загружает очередной выполняемый процесс.
- Метод свопинга влияет на величину временного кванта стратегии RR. Для эффективной загрузки ЦП время свопинга должно быть существенно меньше времени счета. Недостаток «чистого» свопинга заключается в больших потерях времени на загрузку и выгрузку процессов.
- Поэтому в современных ОС используются модифицированные варианты свопинга, например, свопинг включается только в том случае, когда количество процессов памяти становится слишком большим.

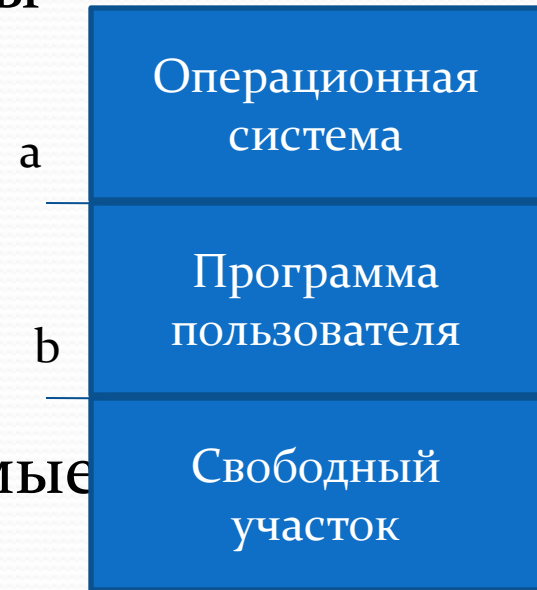
# Методы размещения процессов

- Методы размещения процессов в основной памяти по отношению к расположению участков памяти, выделенной для одной и той же программы, делят на 2 класса:
  - Метод смежного размещения
  - Метод несмежного размещения
- **Смежное размещение** является простейшим и предполагает, что в памяти начиная с некоторого начального адреса, выделяется один непрерывный участок адресного пространства.
- **При несмежном размещении** программа разбивается на множество частей, которые располагаются в различных, необязательно смежных (расположенных рядом) участках адресного пространства.

# В однопрограммном режиме

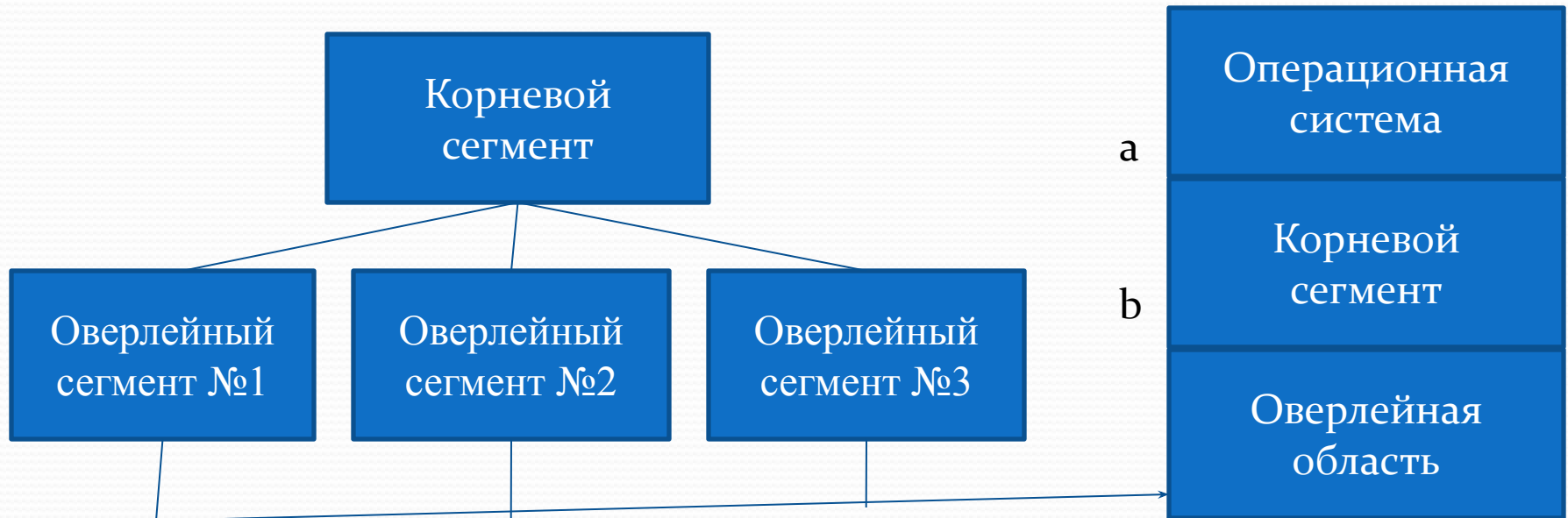
В процессе выполнения программы все ее адреса должны быть  $\geq a$ , иначе произойдет повреждение операционной системы.

Во время работы прикладной программы все адреса, генерируемые ЦП, сравниваются с содержимым регистра границы( $a$ ). Если генерируется адрес  $< a$ , работа программы прерывается.



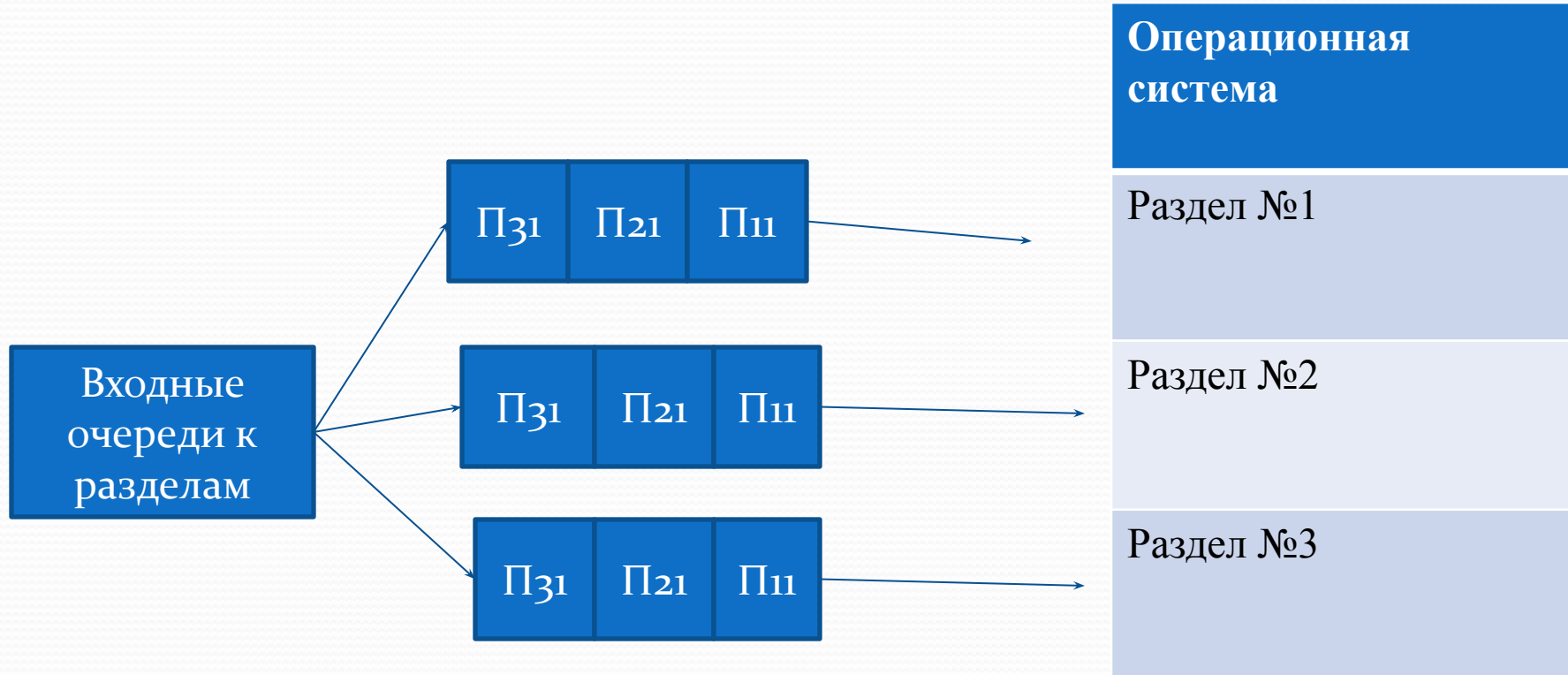
# В однопрограммном режиме

- Для того, чтобы при смежном размещении загружать программы, размеры которых превышают размеры ОЗУ, используют метод **оверлейных сегментов**.



# Мультипрограммирование с фиксированными разделами

(MFT-multiprogramming with a fixed number of tasks)



предполагает разделение адресного пространства на ряд разделов фиксированного размера, в каждом разделе помещается один процесс. Если соответствующий адресам процесса раздел занят, процесс остается в очереди во внешней памяти даже в том случае, когда другие разделы свободны.

# Мультипрограммирование с фиксированными разделами

- Уменьшить фрагментацию памяти при MFT можно, если загрузочные модули создаются в перемещаемых адресах. Такой модуль может быть загружен в любой свободный раздел после соответствующей настройки.
- При мультипрограммировании с трансляцией в перемещаемых адресах имеются 2 причины фрагментации:
- Размер загруженного процесса меньше размера, занимаемого разделом (внутренняя фрагментация)
- Размер процесса в очереди больше размера свободного раздела, и этот раздел остается свободным (внешняя)



# Мультипрограммирование с фиксированными разделами

- Для защиты памяти при МФТ необходимы 2 регистра :
  - регистр верхней границы(наименьший адрес)
  - регистр нижней границы (наибольший адрес).
- Прежде чем программа в разделе N начнет выполняться, ее граничные адреса загружаются в соответствующие регистры.
- В процессе работы программы все формируемые ею адреса контролируются  $a < \text{Адр} < b$ .
- При выходе программы за отведенные ей границы работа программы прерывается.

# Мультипрограммирование с переменными разделами

(MVT-multiprogramming with a variable number of tasks)

- предполагает разделение памяти на разделы и использование загрузочных модулей в перемещаемых адресах, однако **границы разделов не фиксируются.**
- **В начальной фазе отсутствует фрагментация**, связанная с тем, что размер очередного процесса меньше размера, занимаемого этим процессом раздела. На этой фазе причиной фрагментации является несоответствие размера очередного процесса и оставшегося участка памяти. По мере завершения работы программы освобождаются отдельные разделы. В том случае, когда освобождаются смежные разделы, границы между ними удаляются и разделы объединяются.
- За счет объединения и слияния смежных разделов образуются большие фрагменты, в которых можно разместить большие программы из очереди. Таким образом, на фазе повторного размещения действуют те же причины фрагментации, что и для метода MFT.

# Мультипрограммирование с переменными разделами и уплотнением памяти

- может создаться ситуация, когда в памяти образуется множество малых фрагментов, каждый из которых мал для процесса, а в сумме – превышает размер процесса.
- **Уплотнением памяти** называется перемещение всех занятых разделов по адресному пространству памяти таким образом, чтобы свободный фрагмент занимал одну связную область.

# Мультипрограммирование с переменными разделами и уплотнением памяти

- На практике реализация уплотнения памяти сопряжена с усложнением ОС и обладает следующими недостатками:
  - В тех случаях, когда мультипрограммная смесь неоднородна по отношению к размерам программ, возникает необходимость в частом уплотнении, что расходует ресурс процессорного времени и компенсирует экономию ресурса памяти
  - Во время уплотнения все прикладные программы переводятся в состояние ожидания, что приводит к невозможности выполнения программ в реальном времени.

# Основные стратегии заполнения свободного раздела

- В том случае, когда освобождается очередной раздел, ОС должна выбрать один из процессов.
- Алгоритм выбора может использовать одну из следующих стратегий:
  - Стратегия наиболее подходящего – выигрыш в памяти
  - Стратегия первого подходящего
  - Стратегия наименее подходящего – в этом случае остающийся фрагмент часто достаточен для размещения еще одного процесса