


# Тема 2.6. Управление виртуальной памятью

Операционные системы

- 
- Виртуальная память – это технология, которая позволяет выполнять процесс, который может частично располагаться в ОП,
  - т.о. Виртуальная память позволяет выполнять программы, размеры которых превышают размеры физического адресного пространства.

# Перемещение страниц по запросу

- Виртуальная память чаще всего реализуется на базе страничной организации памяти, совмещенной со свопингом страниц. Свопингу подвергаются только те страницы, которые необходимы ЦП.
- Программа может выполняться на ЦП, когда часть страниц в ОП, а часть во внешней
- В процессе выполнения новая страница не перемещается в ОП пока в ней не возникла необходимость

- Для учета распределения страниц между внешней памятью и оперативной каждая строка таблицы страниц дополняется битом местонахождения страницы(valid/ invalid bit)
- Если ЦП пытается использовать недоступную( invalid) страницу, возникает событие – страничная недостаточность(paging fault) – это вызывает прерывание выполнения программы и передачу управления ОС. Реакция ОС – загрузка страницы в ОП.

# Основные этапы обработки страничной недостаточности

- ЦП, прежде чем преобразовать логический адрес в физический, проверяет значение бита
- Если `invalid`, то процесс прерывается и управление передается ОС
- Отыскивается необходимая страница во внешней памяти и свободная страничная рамка в основной
- Загрузка страницы
- Выставляется `valid`
- Управление передается прерванному процессу

# Замещение страниц

В процессе обработки страничной недостаточности ОС может обнаружить, что все страничные рамки заняты и тогда возможны режимы:

- Приостановка прерванного процесса
- Уменьшение на 1 количества процессов для освобождения всех ею занимаемых страничных рамок
- Использования метода замещения страниц

**метод замещения страниц** – в оперативной памяти выбирается наименее важная или используемая страница, которая временно перемещается в пространство свопинга, а на ее место - страница, которой не хватало

# Алгоритм обработки страничной недостаточности

- Определяется местонахождение страницы путем анализа бита местонахождения
- Если `invalid` – разыскивается свободная стр.рамка
- Если есть , то она используется
- Если нет, то используется алгоритм замещения, который выбирает страницу-жертву
- Жертва перемещается в пространство свопинга и таблица страниц редактируется
- Нужная страница помещается на место жертвы и таблица страниц редактируется
- Управление передается прерванному процессу

# Алгоритм обработки страничной недостаточности

- требует двухстраничных перемещений, но жертву можно не копировать, если за время с последнего перемещения, ее содержимое не модифицировалось. Для учета факта **модификации** в таблицу страниц вводится дополнительный **бит**, который меняет свое значение при изменении содержимого таблицы.
- Для практического использования метода обмена страниц по запросу необходимы 2 алгоритма:
- **Алгоритм распределения страничных рамок**
- **Алгоритм замещения страниц**



# Алгоритм распределения страничных рамок

- решает сколько страничных рамок в ОП выделить каждому из процессов мультипрограммной смеси

# Алгоритм замещения страниц

- решает какую из страниц выбрать в качестве жертвы
- Реализуется с помощью:
  - Алгоритма FIFO (first –in–first–out)- замещается наиболее старая страница
  - Оптимального алгоритм
  - Алгоритма LRU (least recently used) – дольше других неиспользуемый
  - Алгоритм LFU (least frequently used) – реже других используемый
  - Случайный (random) выбор

# Алгоритм замещения страниц FIFO (first –in–first–out)

- замещается наиболее старая страница.
- (+)учет времени не обязателен, когда все страницы связаны в FIFO-очередь и каждая помещаемая в память страница ставится в хвост очереди.
- (-)о не учитывается используемость страницы, это может привести к немедленному возвращению только что замещенной страницы.

# Оптимальный алгоритм

- **замещается та страница, на которую нет ссылки на протяжении наиболее длительного периода времени. Для организации этого алгоритма необходимо каждый раз сканировать весь поток ссылок, поэтому он не реализуется на практике и используется для оценки реально работающих алгоритмов**

# Алгоритм LRU (least recently used)

## – дольше других неиспользуемый

- выбирает для замещения ту страницу, на которую не было ссылок на протяжении наиболее длинного периода времени. Он ассоциирует с каждой страницей время последнего использования этой страницы.
  - Подход на основе логических часов (счетчика)
  - Подход на основе стека номеров страниц

# Подход на основе ЛОГИЧЕСКИХ ЧАСОВ

- ассоциирует с каждой строкой таблицы поле «время использования», а в ЦП добавляются логические часы. Логические часы увеличивают свое значение при каждом обращении к памяти. Каждый раз, когда осуществляется ссылка на страницу, значение регистра логических часов копируется в поле «время использования». Заменяется страница с наименьшим значением в отмеченном поле путем сканирования всей таблицы страниц.

# Подход на основе стека номеров страниц

- – стек хранит номера страниц, упорядоченных в соответствии с историей их использования – на вершине – только что использованная страница. Как только осуществляется ссылка на страницу, она перемещается на вершину стека, а остальные сдвигаются вниз.

# Сегментный способ

## организации виртуальной памяти

- И все же естественным способом разбиения программы на части является ее разбиение на логические элементы – так называемые сегменты. Ведь каждая программа обычно состоит из подпрограмм, таким образом, каждая подпрограмма и будет сегментом, которому будет выделена физическая память.



- Вся информация о текущем размещении сегментов задачи в памяти обычно сводится в таблицу сегментов. Каждая задача имеет свою таблицу сегментов. Виртуальный адрес будет состоять из двух полей - номера сегмента и смещения относительно начала сегмента. Если сегмент в данный момент в памяти, то ставится метка - бит присутствия (P). Помимо информации о месторасположении сегмента, в описателе сегмента содержатся данные о его типе (сегмент кода или сегмент данных), правах доступа к сегменту, отметка об обращениях к сегменту. При передаче управления следующей задаче ОС должна занести в соответствующий регистр адрес таблицы описателей сегментов этой задачи. При таком подходе появляется возможность размещать в ОП не все сегменты задачи, а только задействованные в данный момент.