

Управление виртуальной памятью

Подкачка по требованию

Подкачка по требованию

Опр. Подкачка по требованию (demand paging) – метод подкачки, при котором страницы по одной загружаются в оперативную память, только когда процесс явно обращается к ним.

Подкачка по требованию

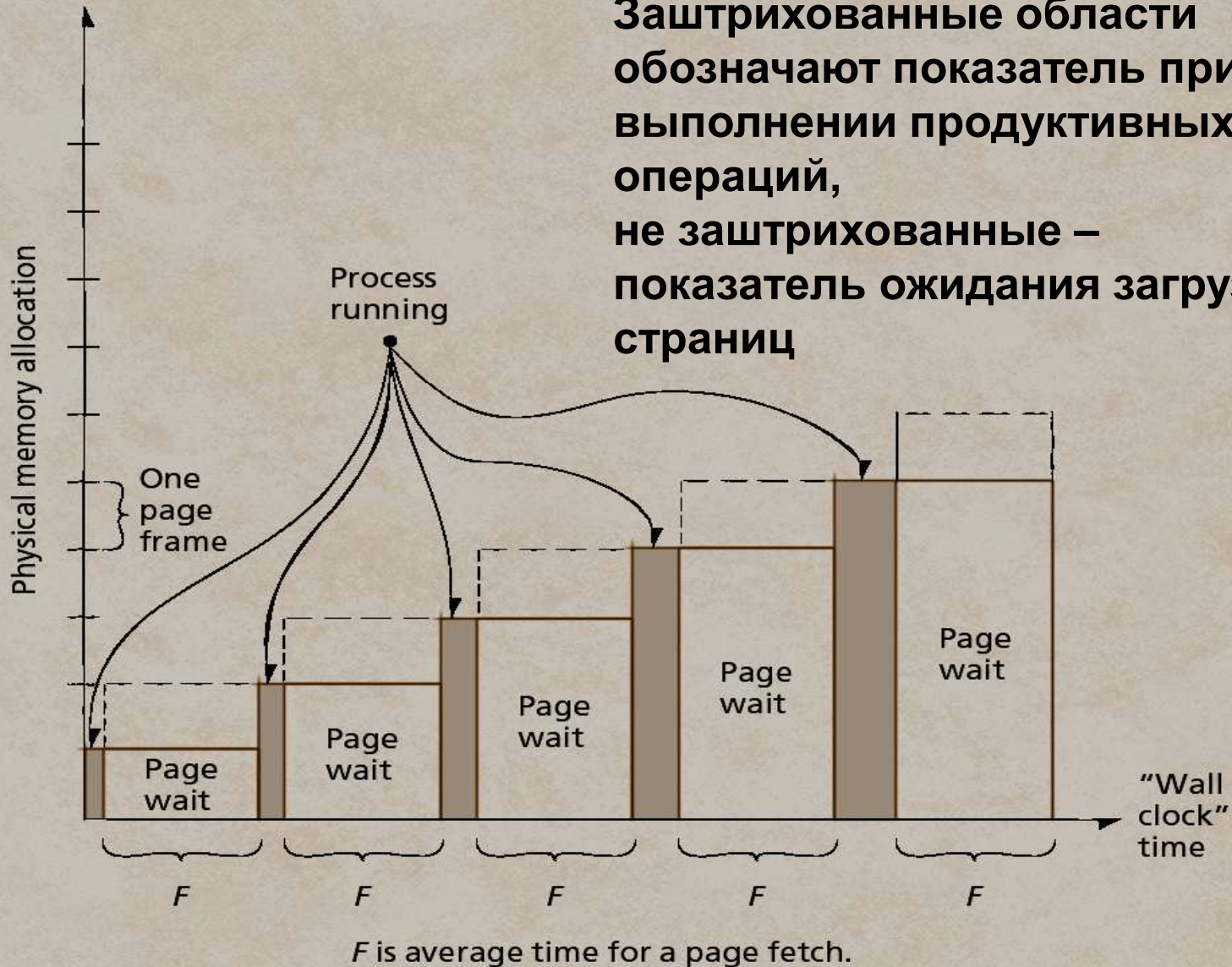
- В начале выполнения процесса система загружает в оперативную память одну страницу, содержащую первую инструкцию процесса
- При обращении к новой странице процесс должен ожидать ее загрузки
- Чем больше страниц процесса уже загружено в оперативную память, тем эта задержка заметнее, так как тем большее пространство в оперативной памяти занимает простаивающий процесс

Пространственно-временной показатель

Опр. Пространственно-временной показатель (space-time product) – значение, равное произведению времени выполнения процесса (т.е. его нахождения в оперативной памяти) и объема оперативной памяти, занятого этим процессом. В идеале стратегии распределения памяти должны уменьшать этот показатель, чтобы повысить степень мультипрограммности.

Пространственно-временной показатель при подкачке по требованию

Заштрихованные области обозначают показатель при выполнении продуктивных операций, не заштрихованные – показатель ожидания загрузки страниц



Вопрос для самопроверки

- Подкачка по требованию может увеличить степень мультипрограммности системы?
(Да/Нет)

Вопрос для самопроверки

- Подкачка по требованию может увеличить степень мультипрограммности системы? (Да/Нет)
- Да. Подкачка по требованию может увеличить степень мультипрограммности, поскольку система будет загружать в память только страницы, действительно требующиеся процессам. Поэтому в физической памяти сможет разместиться большее количество процессов.

Вопрос для самопроверки

- Подкачка по требованию может уменьшить степень мультипрограммности системы? (Да/Нет)

Вопрос для самопроверки

- Подкачка по требованию может уменьшить степень мультипрограммности системы? (Да/Нет)
- Да. Пока операционная система извлекает страницы со вторичного устройства хранения, память, занятая процессом, не используется, и степень мультипрограммности может уменьшиться.

Управление виртуальной памятью

Предварительная подкачка

Предварительная подкачка

Опр. Предварительная подкачка (anticipatory paging, prepaging или prefetching) – метод, загружающий в оперативную память страницы процесса, к которым вероятны обращения в ближайшем будущем. Если система правильно выберет страницы для загрузки, общее время работы процесса может уменьшиться.

Факторы, определяющие успех стратегий предварительной загрузки

- Объем памяти, доступный для размещения предварительно загружаемых страниц
- Количество страниц предварительно загружаемых за один заход
- Алгоритм выбора предварительно загружаемых страниц (с использованием временной или пространственной локальности)

Предварительная подкачка

- Если процесс обращается к странице, которой нет в оперативной памяти, система загружает ее и несколько страниц, расположенных поблизости от нее в виртуальной памяти
- Необходимо компактное размещение на вторичных устройствах страниц, смежных в виртуальном адресном пространстве процесса

Предварительная подкачка

- Разница между временем загрузки нескольких страниц, расположенных рядом на диске, и временем загрузки одной страницы невелика
- Предварительная подкачка может использоваться без заметного увеличения задержек по сравнению с подкачкой по требованию
- Linux загружает сразу 8 смежных страниц

Вопрос для самопроверки

- Может ли стратегия предварительной подкачки Linux оказаться неэффективной? (Да/Нет)

Вопрос для самопроверки

- Может ли стратегия предварительной подкачки Linux оказаться неэффективной? (Да/Нет)
- Да. Если процесс обращается к страницам в случайной последовательности, то Linux, вероятно, будет загружать в память страницы не нужные процессы и засорять ее.

Вопрос для самопроверки

- С учетом локальности подкачка по требованию эффективнее предварительной подкачки? (Да/Нет)

Вопрос для самопроверки

- С учетом локальности подкачка по требованию эффективнее предварительной подкачки? (Да/Нет)
- Нет. Локальность приводит к большей эффективности предварительной подкачки, поскольку операционная система сможет с высокой вероятностью предугадать, какие страницы процесс будет использовать.

Управление виртуальной памятью

Стратегия замены страниц FIFO

Стратегия замены страниц

Опр. Стратегия замены страниц (page-replacement strategy) – стратегия, определяющая, какие страницы убрать из оперативной памяти, чтобы освободить место для требуемых в данный момент. Стратегии замены страниц пытаются оптимизировать производительность за счет предугадывания дальнейшего использования страниц.

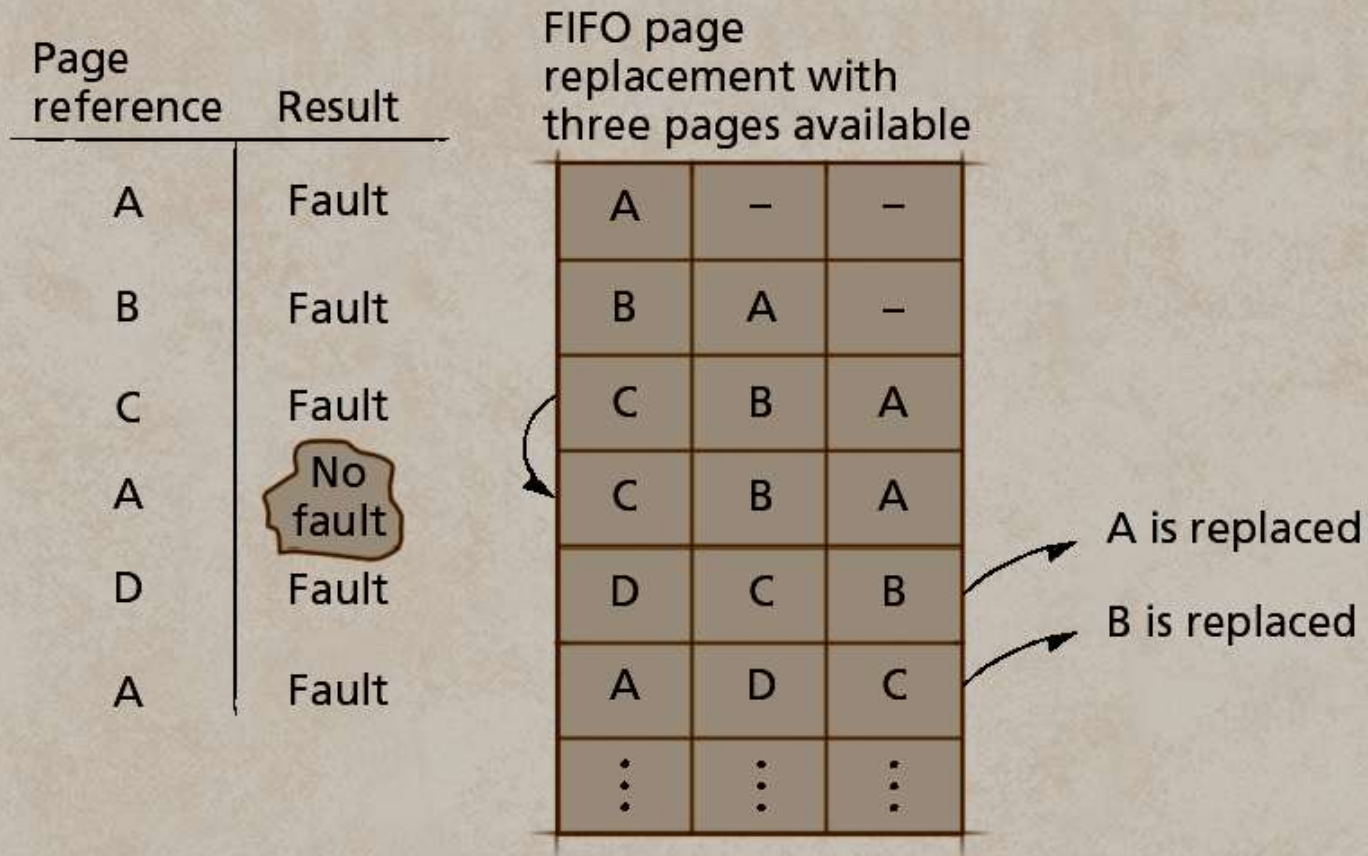
Стратегия замены страниц FIFO

Опр. Стратегия замены страниц FIFO (FIFO page-replacement strategy) – стратегия замены страниц, заменяющая страницу, которая дольше всего находилась в памяти. Эта стратегия связана с низкими накладными расходами, но обычно неточно предсказывает будущие обращения к страницам.

Страничный промах

Опр. Страничный промах (missing-page fault) – ситуация, возникающая при обращении к странице, которая в данный момент не загружена в оперативную память. Если происходит такой промах, операционная система должна загрузить требуемую страницу со вторичного устройства хранения.

Стратегия замены страниц FIFO



Вопрос для самопроверки

- Стратегия замены страниц FIFO обладает высокой производительностью? (Да/Нет)

Вопрос для самопроверки

- Стратегия замены страниц FIFO обладает высокой производительностью? (Да/Нет)
- Нет. Стратегия FIFO заменяет страницы согласно их возрасту, который в отличие от локальности является плохой основой для предсказания будущих обращений к страницам.

Вопрос для самопроверки

- Стратегия замены страниц FIFO редко используется на практике? (Да/Нет)

Вопрос для самопроверки

- Стратегия замены страниц FIFO редко используется на практике? (Да/Нет)
- Да. Но ее модификации служат основой различных практически используемых стратегий замены страниц.

Управление виртуальной памятью

Стратегия замены дольше всего
не использовавшихся страниц
(LRU)

Стратегия замены страниц LRU

Опр. Стратегия замены дольше всего не использовавшихся страниц (Least Recently Used page replacement strategy, LRU) – стратегия замены страниц, заменяющая страницу, к которой дольше всего не обращался процесс. Эта стратегия обычно хорошо предсказывает будущие обращения к страницам, но она связана со значительными накладными расходами.

Стратегия замены страниц LRU

- Можно реализовать с помощью списка, содержащего по одной записи для каждого занятого кадра оперативной памяти
- При каждом обращении к странице памяти LRU будет помещать запись об этой странице в голову списка
- Более старые записи будут отодвигаться к концу списка

Стратегия замены страниц LRU

- Когда нужно заменить страницу на новую, страница, подлежащая замене выбирается из хвоста списка
- Система освобождает соответствующий кадр страницы и помещает в освободившийся кадр новую страницу
- Запись об этом кадре перемещается в голову списка, поскольку к странице в нем обращались последней

Стратегия замены страниц LRU

Page reference	Result	LRU page replacement with three pages available		
A	Fault	A	-	-
B	Fault	B	A	-
C	Fault	C	B	A
B	No fault	B	C	A
B	No fault	B	C	A
A	No fault	A	B	C
D	Fault	D	A	B
A	No fault	A	D	B
B	No fault	B	A	D
F	Fault	F	B	A
B	No fault	B	F	A

Вопрос для самопроверки

- Стратегия LRU повышает скорость выполнения процессов с пространственной локальностью?
(Да/Нет)

Вопрос для самопроверки

- Стратегия LRU повышает скорость выполнения процессов с пространственной локальностью?
(Да/Нет)
- Нет. Эта стратегия повышает скорость выполнения процессов с временной локальностью.

Вопрос для самопроверки

- Стратегия LRU редко используется на практике? (Да/Нет)

Вопрос для самопроверки

- Стратегия LRU редко используется на практике? (Да/Нет)
- Да. Потому что она требует значительных накладных расходов на поддержание упорядоченного списка страниц и его переупорядочивание.

Управление виртуальной памятью

Стратегия замены давно не
используемых страниц (NUR)

Бит изменения

Опр. Бит изменения (modified bit) – поле в записи страничной таблицы, значение которого указывает, изменялось ли содержимое страницы. Если этот бит установлен, то перед заменой этой страницы на новую ее содержимое нужно скопировать на вторичное запоминающее устройство.

Бит обращения

Опр. Бит обращения (referenced bit) – поле в записи страничной таблицы, значение которого указывает были ли обращения к странице. Если этот бит установлен, это означает, что обращения были.

Стратегия замены страниц NUR

Опр. Стратегия замены давно не используемых страниц (Not Used Recently page replacement strategy, NUR) – аппроксимация стратегии замены страниц LRU, требующая меньших накладных расходов при использовании.

Стратегия замены страниц NUR

- В зависимости от значений битов изменения и обращения все страницы делятся на четыре группы

<i>Group</i>	<i>Referenced</i>	<i>Modified</i>	<i>Description</i>
Group 1	0	0	Best choice to replace
Group 2	0	1	[Seems unrealistic]
Group 3	1	0	
Group 4	1	1	Worst choice to replace

Стратегия замены страниц NUR

- Страницы группы 1 – лучшие кандидаты на замену, а группы 4 – худшие

<i>Group</i>	<i>Referenced</i>	<i>Modified</i>	<i>Description</i>
Group 1	0	0	Best choice to replace
Group 2	0	1	[Seems unrealistic]
Group 3	1	0	
Group 4	1	1	Worst choice to replace

Стратегия замены страниц NUR

- При наличии страниц в группе с меньшим номером, страница для замены выбирается из этой группы случайным образом

<i>Group</i>	<i>Referenced</i>	<i>Modified</i>	<i>Description</i>
Group 1	0	0	Best choice to replace
Group 2	0	1	[Seems unrealistic]
Group 3	1	0	
Group 4	1	1	Worst choice to replace

Стратегия замены страниц NUR

- Группа 2 измененных страниц, к которым не было обращений возникает потому, что NUR периодически сбрасывает биты обращений для всех страниц (при интенсивной работе системы они все через некоторое время становятся равными 1)

<i>Group</i>	<i>Referenced</i>	<i>Modified</i>	<i>Description</i>
Group 1	0	0	Best choice to replace
Group 2	0	1	[Seems unrealistic]
Group 3	1	0	
Group 4	1	1	Worst choice to replace

Вопрос для самопроверки

- Бит изменения повышает производительность стратегии NUR? (Да/Нет)

Вопрос для самопроверки

- Бит изменения повышает производительность стратегии NUR? (Да/Нет)
- Да. Бит изменения позволяет операционной системе определить, какие страницы можно заменять без предварительного сброса их на диск. Выбор для замены страниц неизменного содержимого позволяет уменьшить число операций ввода/вывода при замене страниц.

Вопрос для самопроверки

- Может NUR заменить хуже всего подходящую для замены страницу?
(Да/Нет)

Вопрос для самопроверки

- Может NUR заменить хуже всего подходящую для замены страницу?
(Да/Нет)
- Да. Если бит обращения страницы, к которой сейчас будут обращения, был сброшен непосредственно перед моментом, когда нужно заменить страницу.

Вопрос для самопроверки

- Может ли оказаться, что содержимое страницы изменялось, а обращений к ней не было? (Да/Нет)

Вопрос для самопроверки

- Может ли оказаться, что содержимое страницы изменялось, а обращений к ней не было? (Да/Нет)
- Да. Такое может быть в стратегии NUR. На самом деле обращения к этой странице были, но биты обращений периодически сбрасываются в ноль.

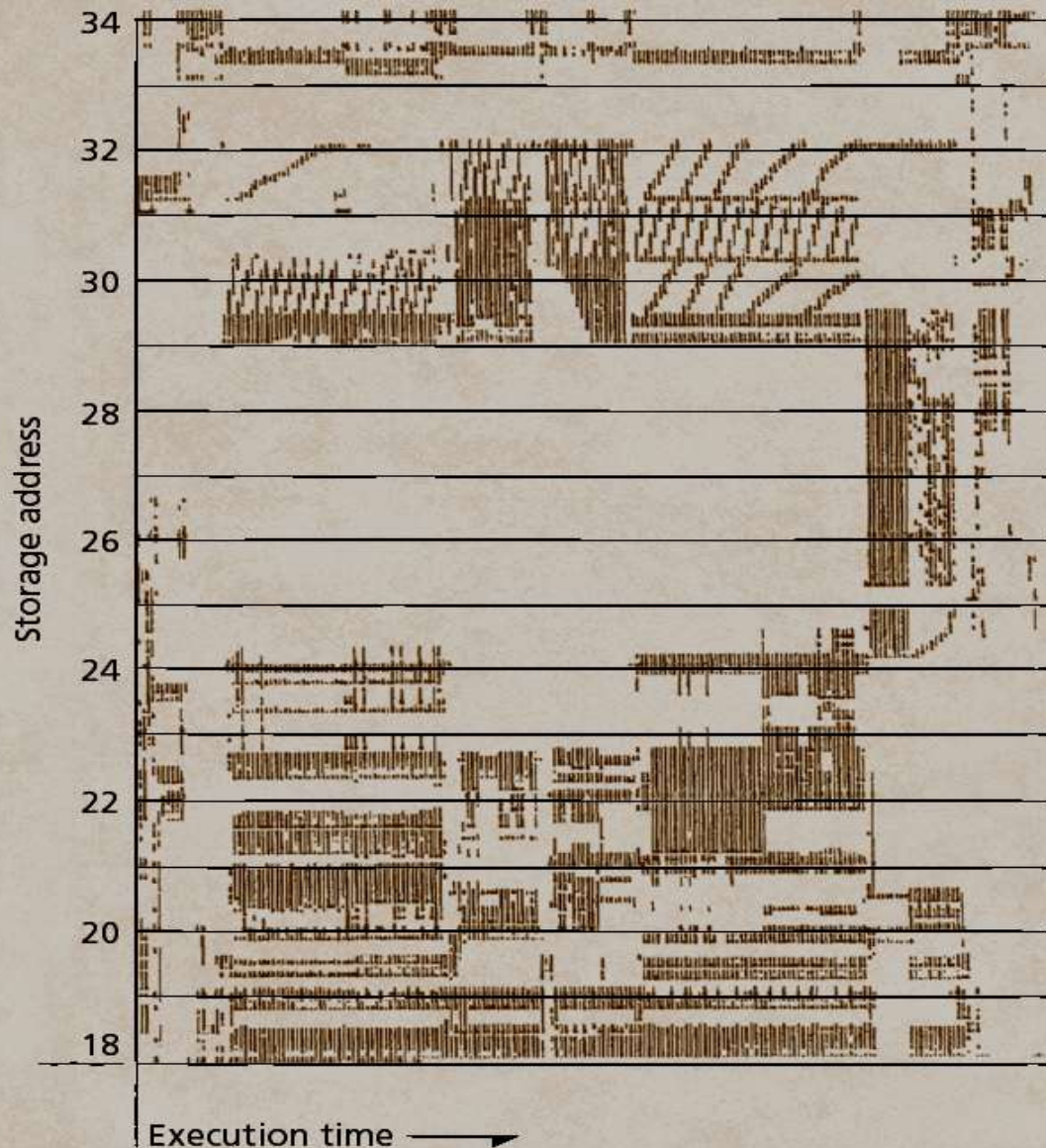
Управление виртуальной памятью

Замена страниц в Linux

Рабочий набор

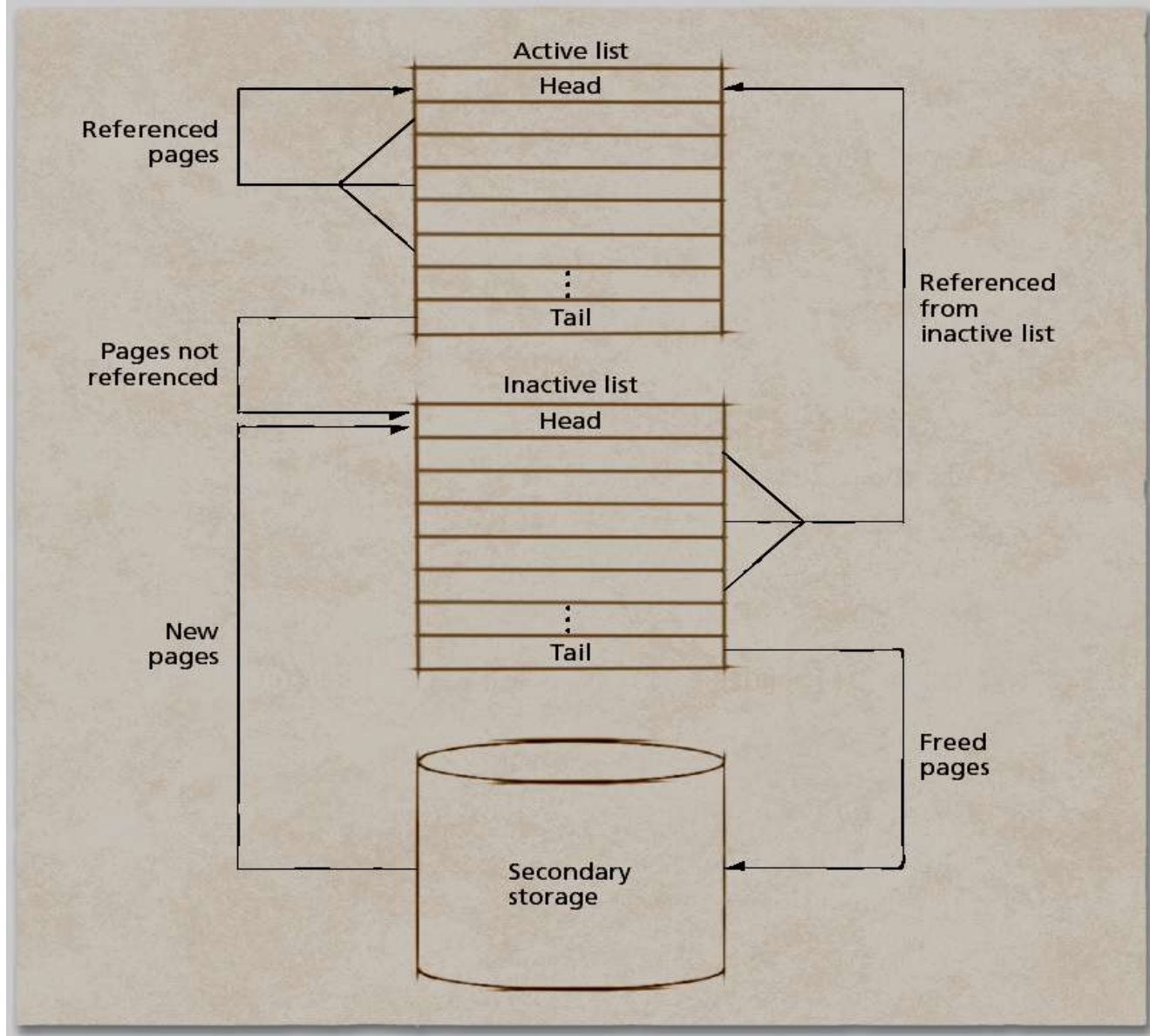
Опр. Рабочий набор (working set) – набор часто используемых процессом страниц виртуальной памяти. Практические исследования показывают, что программы часто проявляют пространственную локальность, обращаясь на разных фазах выполнения к определенным рабочим наборам.

Последовательность обращений к памяти, проявляющая локальность

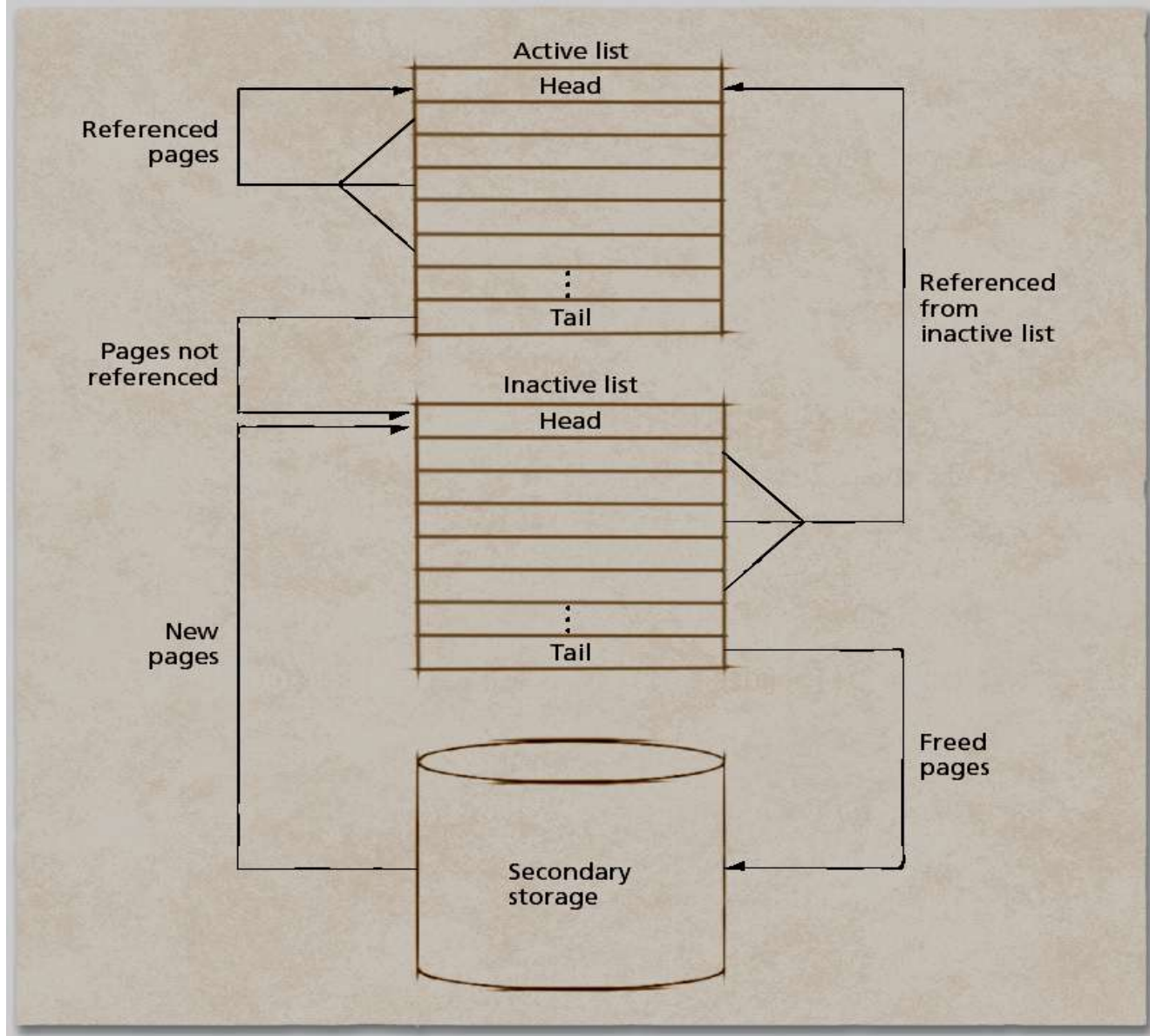


Замена страниц в Linux

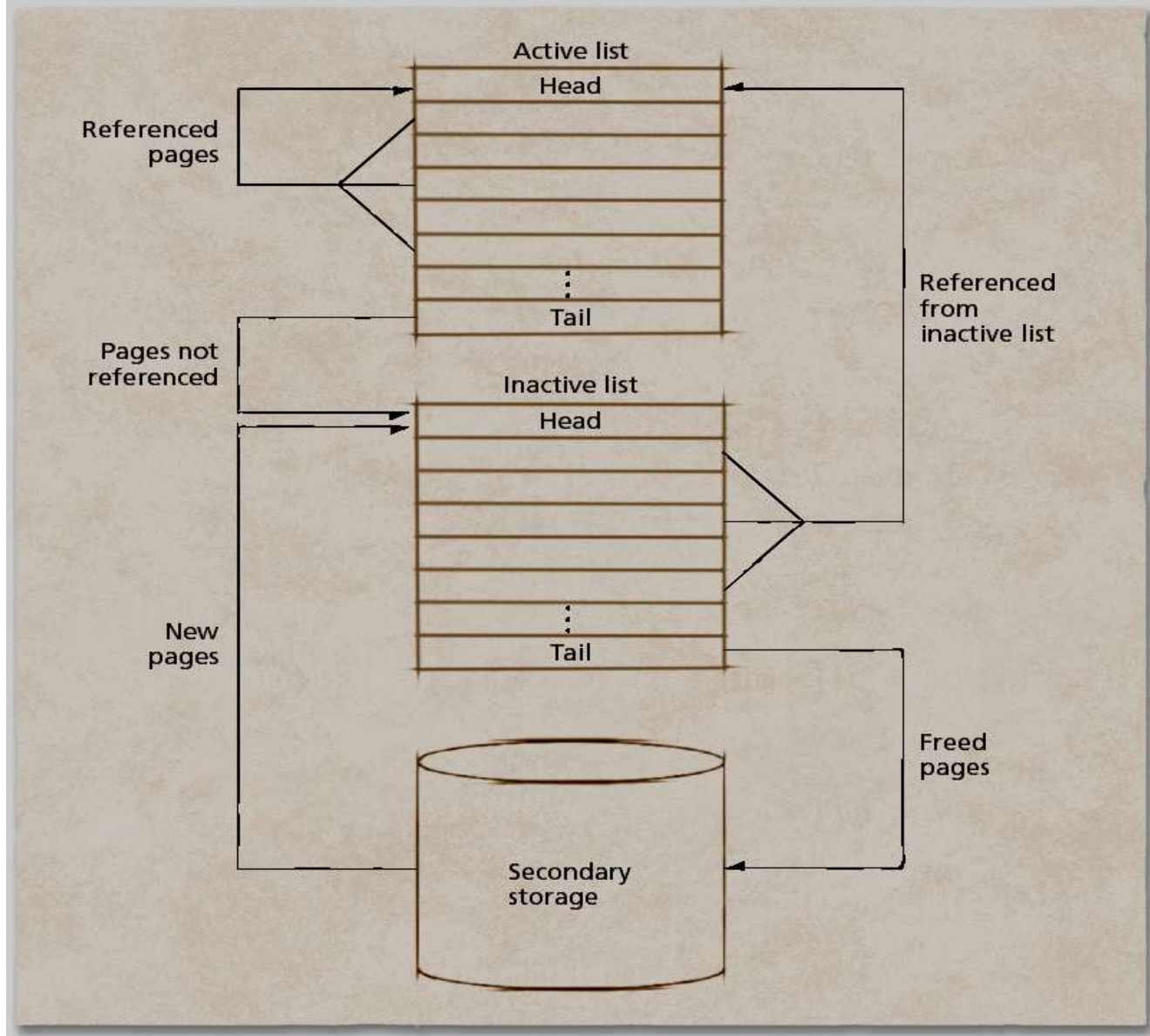
- Страницы делятся на активные и не активные
- Чтобы считаться активной страница должна недавно получить обращение
- Одна из задач системы управления памятью – хранить текущий рабочий набор в активных страницах



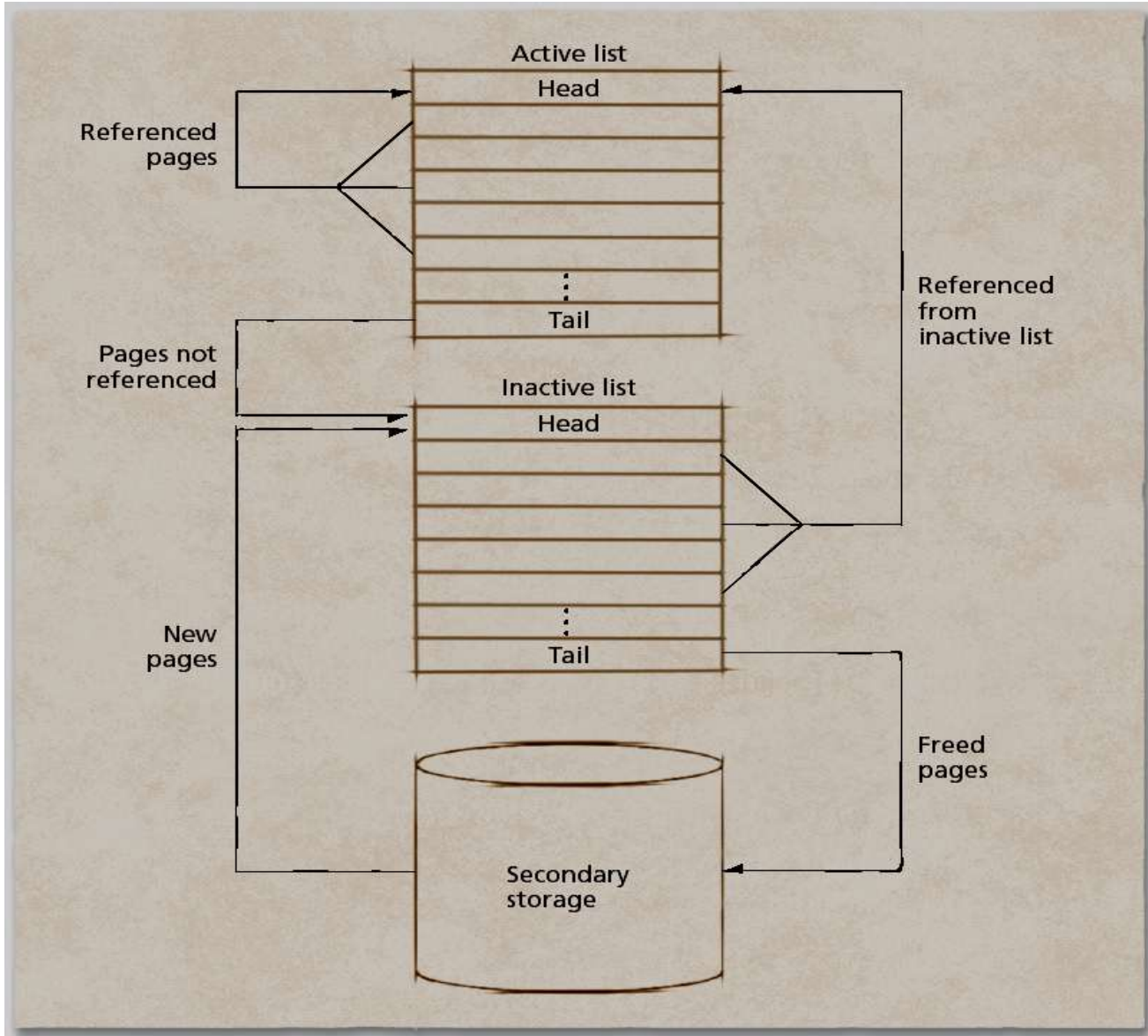
- **Страницы, использовавшиеся позже всего, находятся поблизости от головы активного списка**



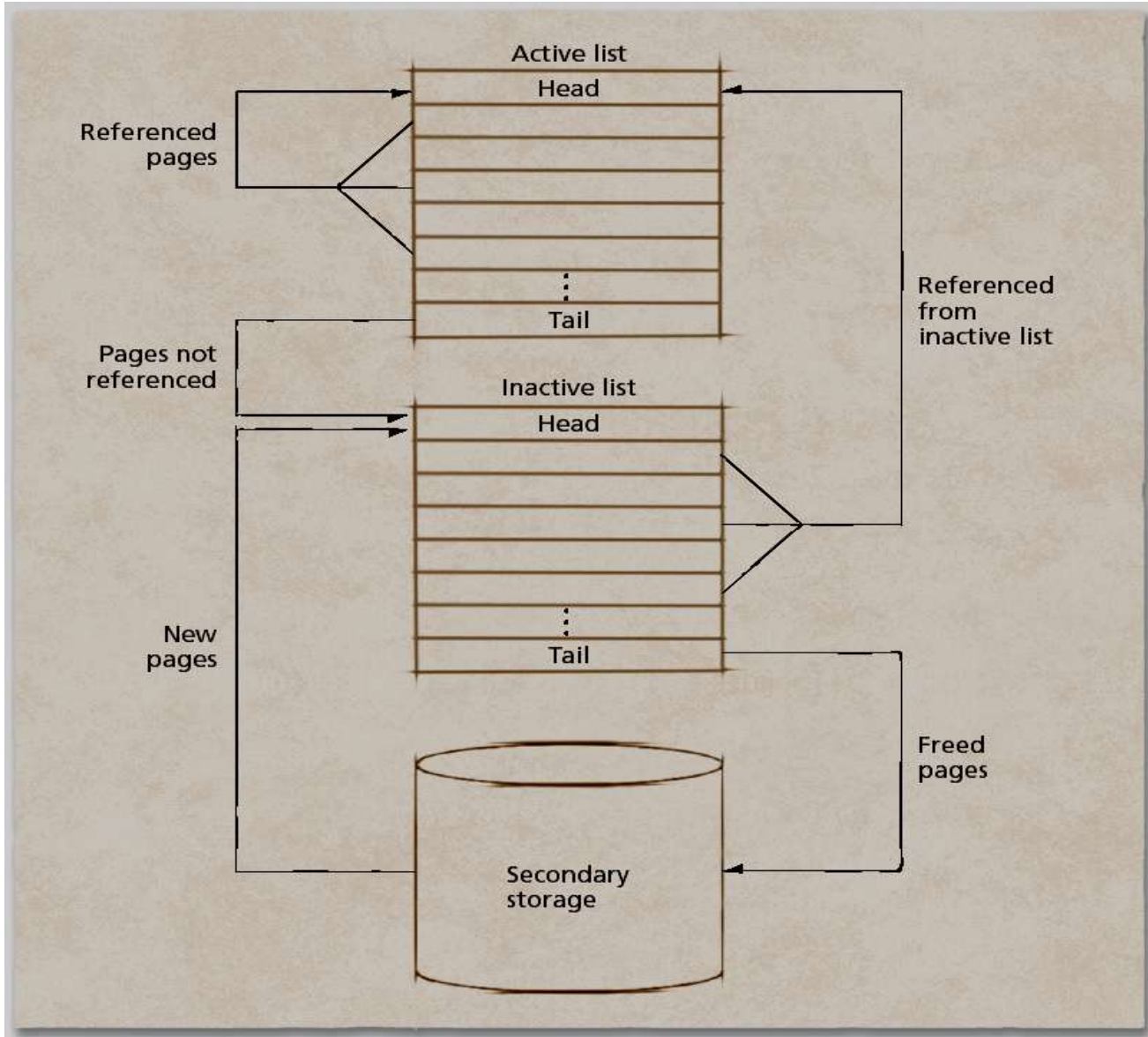
- Страницы, дольше всего не использовавшиеся находятся поблизости от хвоста неактивного списка



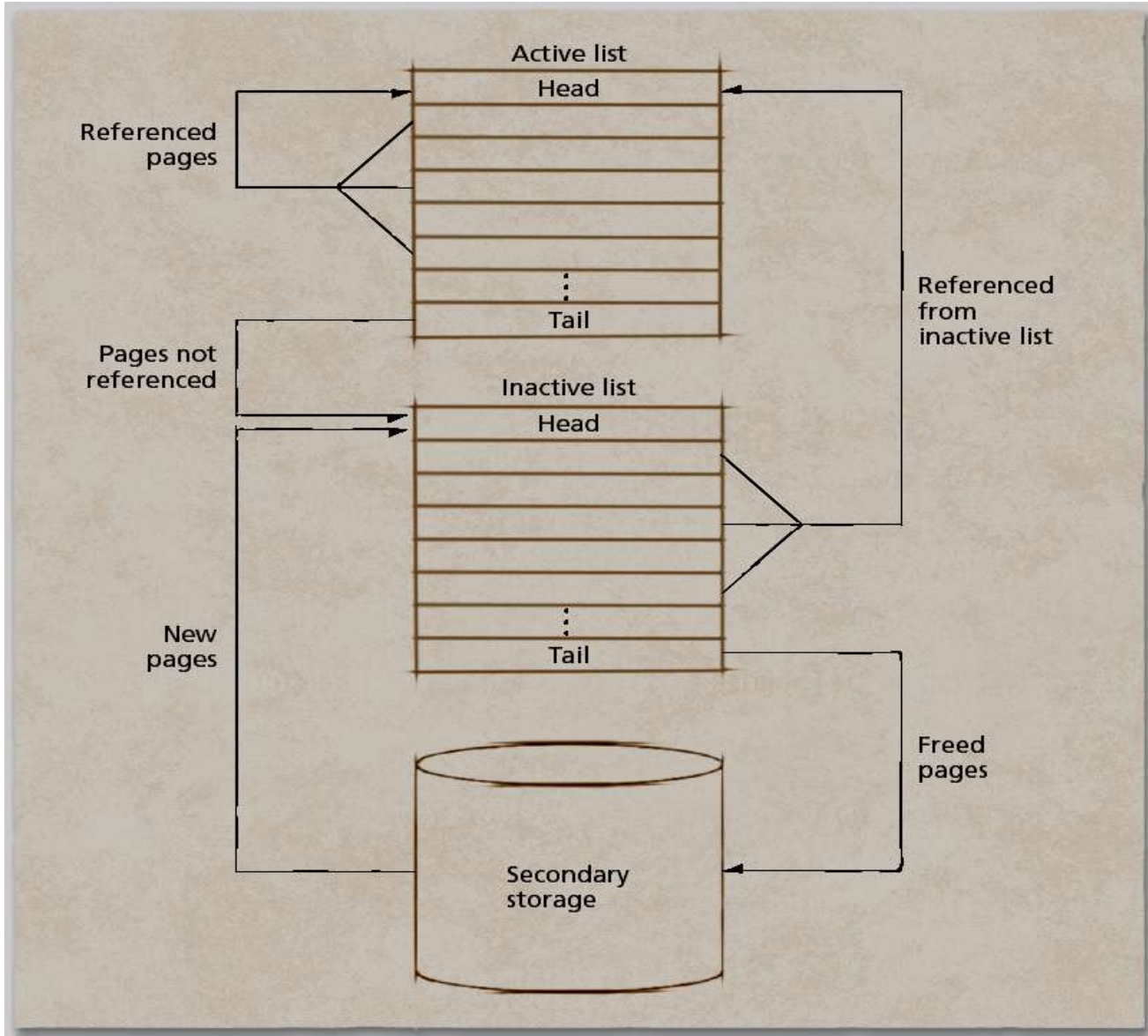
- Когда страница впервые загружается в оперативную память, она помещается в неактивный список и ее бит обращения устанавливается в 1



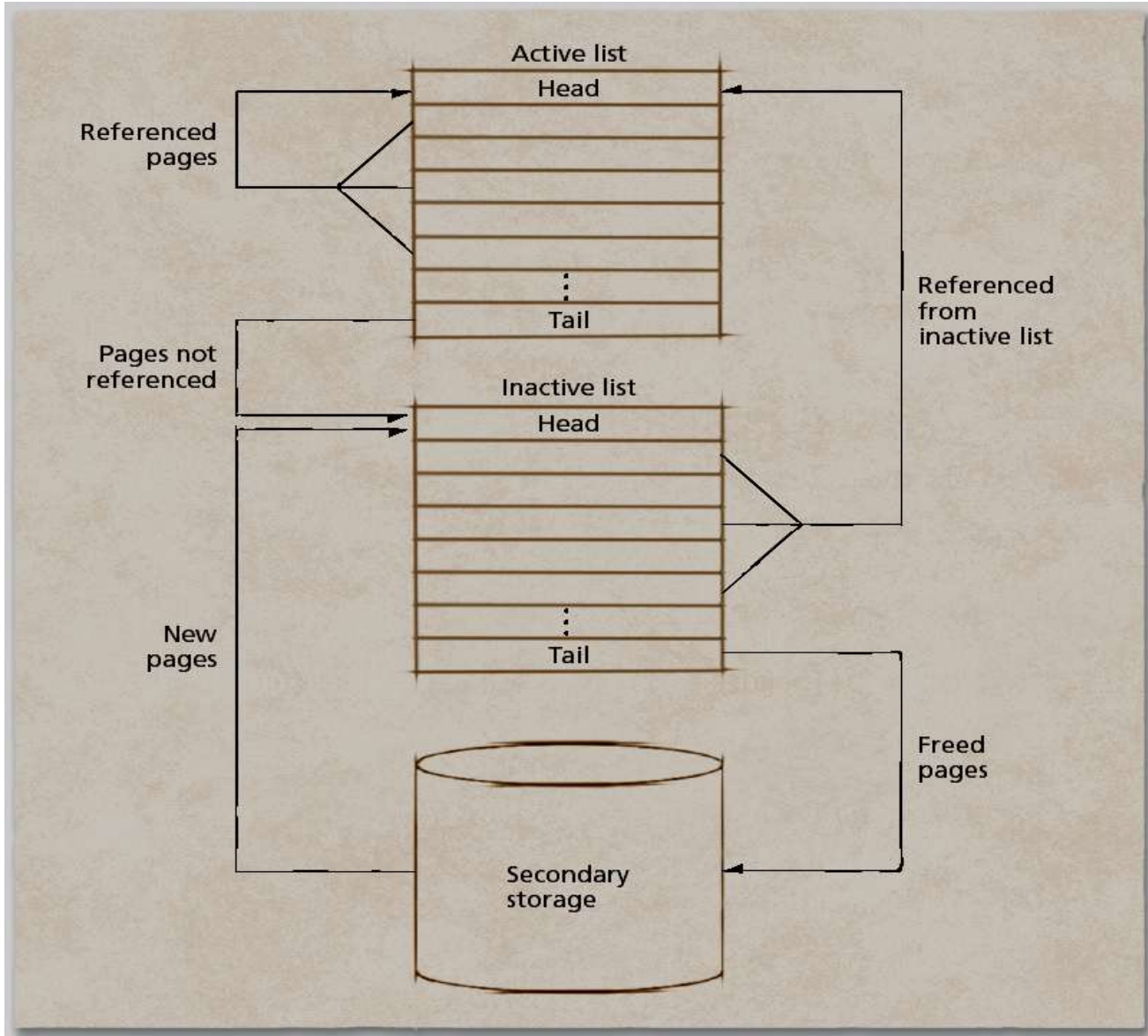
- Если страница неактивна и ее бит обращения уже установлен в 1, а к ней поступает обращение, то диспетчер переносит ее в голову активного списка и сбрасывает ее бит обращения



- Страницы из активного списка, к которым поступают обращения, переносятся в голову этого списка, их бит обращения устанавливается в 1



- **Страницы из хвоста активного списка периодически переносятся в голову неактивного списка; пока страницы находятся в активном списке их нельзя заменить**



- Для замены выбираются страницы из хвоста неактивного списка

Вопрос для самопроверки

- Верно ли, что в активном списке Linux находятся страницы, которым было более чем одно обращение? (Да/Нет)

Вопрос для самопроверки

- Верно ли, что в активном списке Linux находятся страницы, которым было более чем одно обращение? (Да/Нет)
- Да. В активный список страницы попадают из неактивного, следовательно, как минимум после второго обращения к ним.

Вопрос для самопроверки

- Можно ли заменить страницу из активного списка Linux? (Да/Нет)

Вопрос для самопроверки

- Можно ли заменить страницу из активного списка Linux? (Да/Нет)
- Нет. Для замены выбираются страницы из хвоста неактивного списка.

Вопрос для самопроверки

- Всегда ли при обращении страница из неактивного списка Linux переносится в активный? (Да/Нет)

Вопрос для самопроверки

- Всегда ли при обращении страница из неактивного списка Linux переносится в активный? (Да/Нет)
- Нет. Если страница неактивна и ее бит обращения равен 0, а к ней поступает обращение, то диспетчер переносит ее в голову неактивного списка и устанавливает ее бит обращения в 1. Это гарантирует, что недавно использовавшиеся страницы не будут выбраны для замены.

Управление виртуальной памятью

Размер страниц

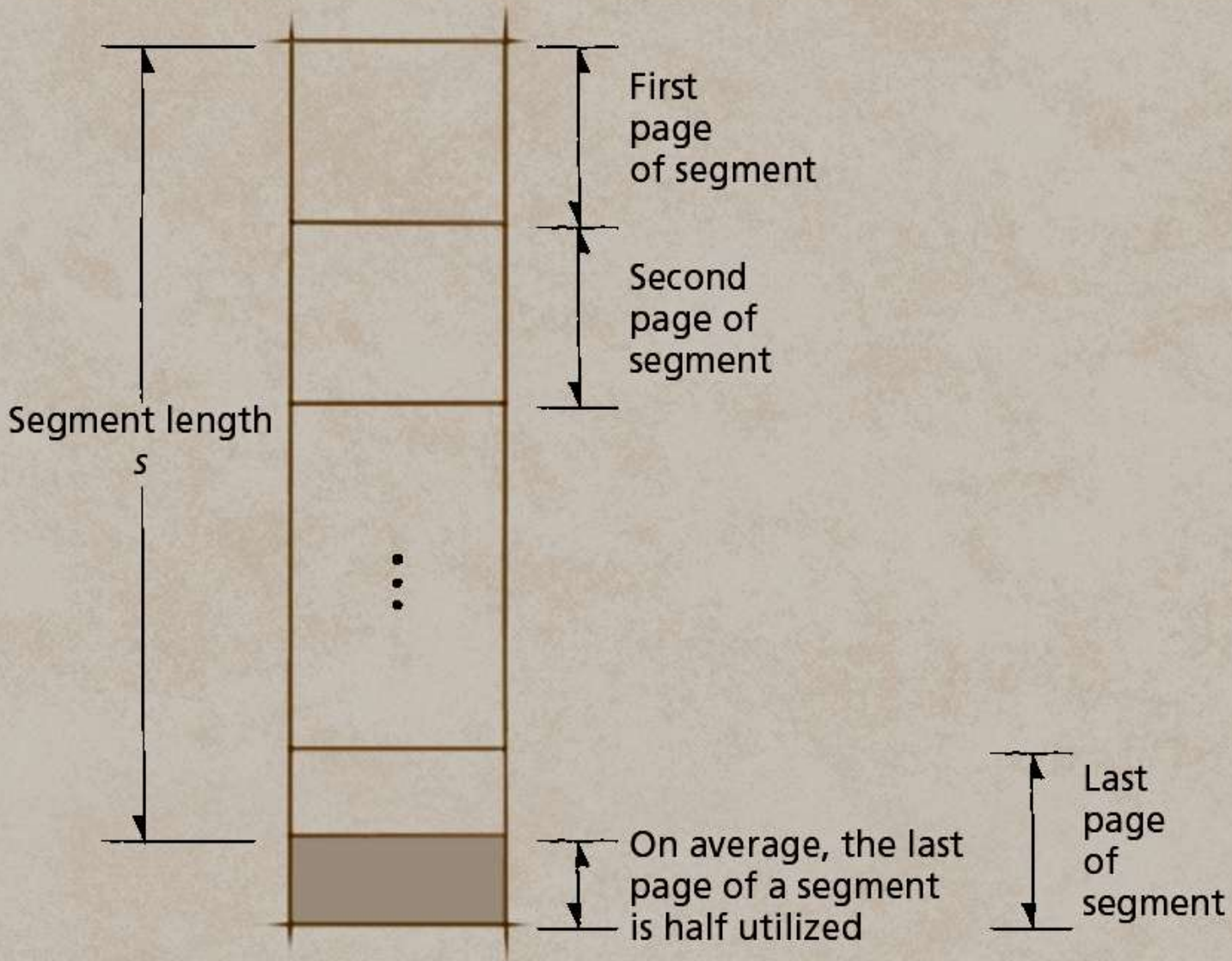
Аргументы за большие страницы

- Чем больше страницы, тем их меньше
- Меньший размер имеют страничные таблицы
- Больше вероятность попадания записи о странице в буфер быстрой трансляции TLB
- Необходимо меньше операций ввода/вывода при замене страниц

Аргументы за маленькие страницы

- Страницы меньшего размера могут позволить процессу создать более компактный рабочий набор
- Чем меньше страницы, тем меньше внутренняя сегментация
- В сегментно-страничных системах каждый сегмент, в среднем, расходует половину одной страницы на внутреннюю фрагментацию

Внутренняя фрагментация в сегментно-страничных системах



<i>Manufacturer</i>	<i>Model</i>	<i>Page Size</i>	<i>Real address size</i>
Honeywell	Multics	1KB	36 bits
IBM	370/168	4KB	32 bits
DEC	PDP-10 and PDP-20	512 bytes	36 bits
DEC	VAX 8800	512 bytes	32 bits
Intel	80386	4KB	32 bits
Intel / AMD	Pentium 4 / Athlon XP	4KB or 4MB	32- or 36 bits
Sun	UltraSparc II	8KB, 64KB, 512KB, 4MB	44 bits
AMD	Opteron / Athlon 64	4KB, 2MB and 4MB	32, 40, or 52 bits
Intel-HP	Itanium, Itanium 2	4KB, 8KB, 16KB, 64KB, 256KB, 1MB, 4MB, 16MB, 64MB, 256MB	Between 32 and 63 bits
IBM	PowerPC 970	4KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB, 64MB, 128MB, 256MB	32 or 64 bits

Вопрос для самопроверки

- В современных системах используются страницы большого размера? (Да/Нет)

Вопрос для самопроверки

- В современных системах используются страницы большого размера? (Да/Нет)
- Да. Память дешевеет и появляется возможность использовать страницы большого размера.

Вопрос для самопроверки

- В современных системах используются страницы небольшого размера?
(Да/Нет)

Вопрос для самопроверки

- В современных системах используются страницы небольшого размера? (Да/Нет)
- Да. Поддержка страниц разных размеров позволяет эффективнее использовать память ценой дополнительных затрат на ее распределение. Например, необходимо иметь столько буферов быстрой трансляции TLB сколько размеров страниц поддерживает система.