

8. Двухуровневая страничная организация

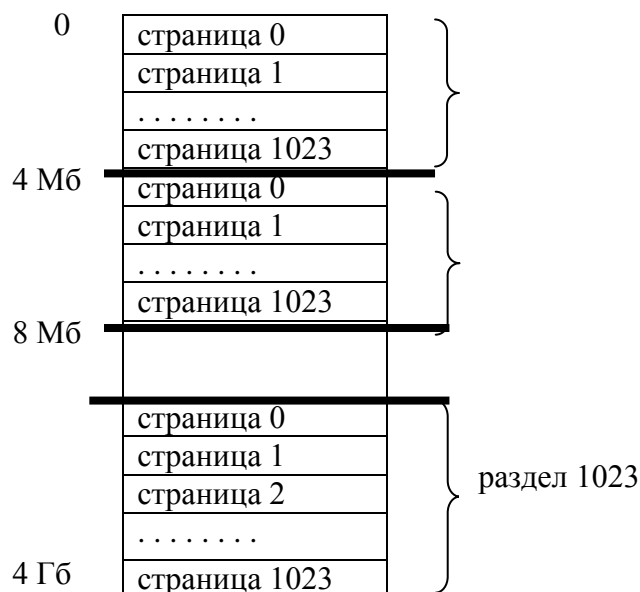
Двухуровневая страничная организация вводится для устранения одного из недостатков простой страничной организации – больших затрат памяти на хранение таблиц страниц процессов. Правда, платой за это является некоторое усложнение преобразования ВА в ФА. Основная идея двухуровневой организации совершенно очевидна: разбить **одну** очень **большую** таблицу страниц на **множество** более **мелких**.

Для этого ВАП разбивается на достаточно крупные разделы, а разделы – на обычные страницы. Все разделы нумеруются от 0 до некоторого М, а страницы внутри раздела - от 0 до N. Виртуальный адрес будет состоять не из двух, а уже из **трех** частей: номер раздела, номер страницы в разделе, смещение на странице. Конкретное число разделов в ВАП и число страниц в разделе зависят от архитектуры базового процессора и типа ОС. Например, 32-х разрядный адрес можно разбить следующим образом:

- старшие 10 битов – номер раздела в ВАП (всего разделов будет 1024);
- средние 10 битов – номер страницы в разделе (число страниц тоже 1024);
- младшие 12 битов – как обычно, смещение адреса на странице (от 0 до 4095).

номер раздела	номер страницы	смещение
31	22 21	12 11 0

Тем самым, все ВАП структурируется следующим образом:



В этом случае для каждого процесса надо создать **одну** таблицу разделов и **1024** таблицы страниц отдельно для каждого раздела. Таблица разделов содержит записи-дескрипторы с информацией о каждом разделе, а каждая таблица страниц, как обычно – записи-дескрипторы для каждой страницы. При этом возникает следующая интересная особенность. Если одна запись в таблице страниц содержит 4 байта (этого вполне достаточно для хранения номера физической страницы и дополнительной управляющей информации), то на всю таблицу страниц надо 4 Кбайта памяти, т.е. ровно **ОДНУ** страницу памяти! Для хранения всех таблиц страниц процесса требуется 1024 страницы, но **большую** их часть можно выгрузить на диск, освободив место для других целей. А вот относительно небольшую таблицу разделов (тоже примерно 4 Кб) надо постоянно хранить в основной памяти. Начальный адрес этой таблицы для активного в данный момент процесса хранится в специальном регистре и изменяется при переключении процессов.

Каждая запись-дескриптор таблицы разделов содержит следующие данные:

- номер физической страницы, в которую загружена таблица страниц для данного раздела;
- признак присутствия этой страницы в памяти;

- дополнительная управляющая информация, аналогичная обычной страничной организации.

Схематично взаимосвязь таблицы разделов и таблиц страниц можно представить следующим образом.

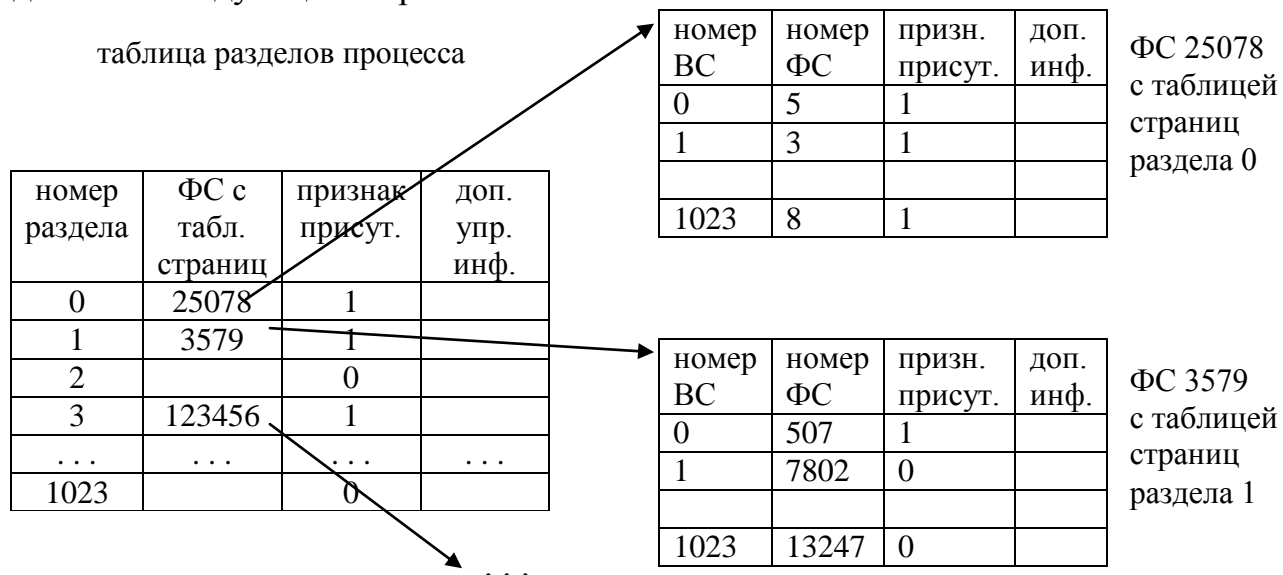
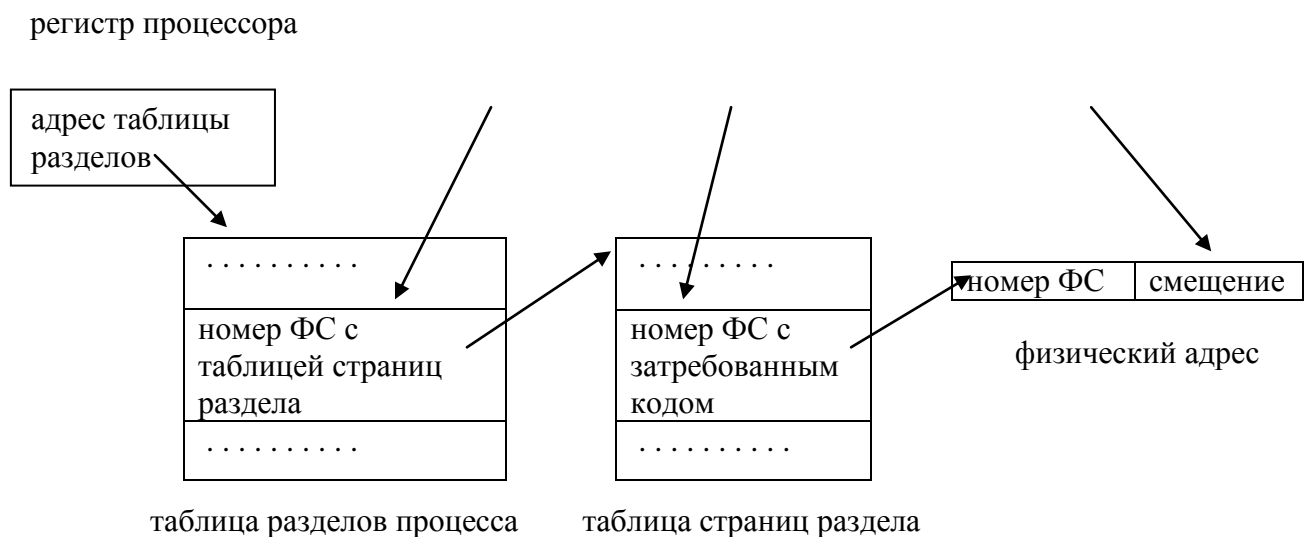


Таблица разделов и необходимые таблицы страниц создаются подсистемой управления памятью при создании процесса, при этом некоторые таблицы страниц могут быть вытеснены на диск. Алгоритм преобразования ВА в ФА выглядит так:

- Этап 1 – обработка таблицы разделов:
 - из ВА выделяется номер раздела и выполняется вход в таблицу разделов;
 - в записи-дескрипторе проверяется признак присутствия страницы с таблицей страниц в памяти;
 - если эта страница отсутствует в памяти, генерируется страничное прерывание, на диске отыскивается запрошенная страница, выполняется ее загрузка в память с вытеснением при необходимости какой-нибудь страницы на диск и установкой признаков присутствия.
- Этап 2 – обработка таблицы страниц:

- из ВА выделяется номер затребованной процессором виртуальной страницы, по которому выполняется вход в найденную на 1 этапе таблицу страниц;
- анализируется запись-дескриптор для затребованной страницы и проверяется ее присутствие в памяти;
- если страницы нет в памяти, то с помощью еще одного страничного прерывания выполняются ее поиск и загрузка в ОП с вытеснением при необходимости одной из страниц;
- из таблицы страниц извлекается номер ФС, который вместе со смещением определяет искомый ФА.

Схематично это можно представить следующим образом:



Из алгоритма видно, что двухуровневая схема требует большего числа операций для преобразования адресов, поэтому большое значение для ее эффективной реализации имеет выполнение некоторых операций на аппаратном уровне. В принципе, программная реализация необходима в основном для обработки страничных прерываний, а собственно само преобразование адресов можно реализовать с помощью логических схем. Кроме того, многие процессоры позволяют еще более ускорить это преобразование за счет сохранения часто используемых дескрипторов страниц в сверхбыстрой кэш-памяти (более подробно использование кэш-памяти рассматривается далее в пособии).

В заключение еще раз отметим достоинства и недостатки двухуровневой страничной организации памяти. Достоинства:

- возможность быстрого формирования ФА только с помощью поразрядных операций;
- полное отсутствие фрагментации основной памяти за счет выделения ее процессам небольшими страницами ;
- относительно небольшие накладные расходы на поддержку необходимых структур данных (таблиц разделов и таблиц страниц) с возможностью вытеснения их на диск.

Недостатки:

- сохраняется жесткое фиксированное разделение кода и данных программы на разделы и страницы без учета логической структуры программы;
- возможное большее число страничных прерываний;
- достаточно сложная реализация механизма общей (разделяемой) памяти для разных процессов.

Частично недостатки страничной организации устраняются за счет использования сегментов, правда – за счет появления своих недостатков.