

К преимуществам сегментной организации в литературе обычно относят следующие.

- Легко можно указать режим доступа к сегменту в зависимости от смысла его данных. Например, сегмент кода программы обычно должен быть доступен только для чтения, а сегмент данных может быть доступен и для записи.
- В том случае, если программа работает с двумя или более структурами данных, каждая из которых может увеличиваться в размерах независимо от других, выделение отдельного сегмента для каждой структуры позволяет освободить программиста от забот, связанных с размещением структур в имеющейся памяти (эти проблемы перекладываются на ОС, которая обязана будет найти место в физической памяти для увеличивающихся сегментов).
- Гораздо реже называется еще одна, более прозаическая причина использования сегментов, которая на самом деле в определенный период являлась очень веской. Если в используемой архитектуре компьютера разрядность адреса в командах слишком мала (например, 16 разрядов, как у процессоров i286, что позволяет адресовать всего лишь 64 Кб), а размер программы и ее данных достигает многих мегабайт, то единственное решение – использовать много сегментов по 64 Кб.

Для современных процессоров разрядность адреса составляет 32 или даже 64 бита, что снимает необходимость возиться с большим количеством мелких сегментов. При этом на первый план выходят достоинства страничной организации:

- программист не должен вообще думать о разбиении программы и ее данных на части ограниченного размера (сегменты), в его распоряжении единое пространство виртуальных адресов;
- исключается возможность фрагментации физической памяти и связанные с этим проблемы;
- как правило, уменьшается обмен данными с диском, поскольку в него включаются только отдельные страницы, а не целые сегменты.

Для сравнительной оценки сегментной и страничной организации полезно также вспомнить историю развития версий Windows. Версия Windows 2.0 была ориентирована на процессор i286, имевший сегментную организацию памяти с 16-разрядным смещением в сегменте. В эти годы фирмы Intel и Microsoft активно защищали сегментную модель, подчеркивая ее достоинства. Однако в Windows 3.0 были уже частично использованы новые возможности процессора i386, а именно, страничная организация памяти. Поскольку эта версия по-прежнему была основана на 16-разрядных адресах, использование сегментов оставалось необходимым, что привело к сложной сегментно-страничной модели памяти. Зато переход к 32-разрядным версиям WindowsNT и Windows 95 сопровождался фактическим отказом от использования сегментного механизма в пользу чисто страничной организации памяти. Формально же теперь все адресное пространство пользователя укладывается в один очень большой сегмент размером 4 Гб.

Большим преимуществом использования виртуальной памяти, как в сегментном, так и в страничном варианте, является возможность легко и просто изолировать процессы в памяти. Для этого достаточно, чтобы система не отображала никакие виртуальные страницы двух разных процессов на одну и ту же физическую страницу. Тогда процессы просто «не будут видеть» друг друга в памяти и не смогут повредить друг другу.

С другой стороны, в некоторых ситуациях желательно, чтобы два или более процессов имели доступ к общей области памяти. Это дает, например, возможность хранить в памяти единственный экземпляр системных библиотек, которым могут пользоваться несколько процессов. Для создания общей памяти достаточно, чтобы виртуальные страницы всех заинтересованных процессов отображались на одни и те же страницы физической памяти.