

Лекция 7. Простейшие схемы управления памятью

Управление памятью

Иерархия памяти

Существует как минимум 2 вида памяти: оперативная и вторичная (дисковая). Оперативная память быстрее, но дороже, чем дисковая.

Из-за внедрения более быстрой памяти мы получаем ускорение всей системы в целом, даже несмотря на то, что оперативная память обладает меньшим объемом.

Принцип локальности

Принцип пространственной локальности. программа всегда работает с небольшим количеством локальных переменных и постоянно переходит из одной локали в другую.

Проблема разрешения адресов

Проблема разрешения адресов -- это проблема о том, что человеку удобнее мыслить в символьных пространствах, а компьютер умеет мыслить только в числах, соответствующих адресным переменным. Исходный текст программы проходит несколько этапов, прежде чем стать чем-то что операционная система может выполнить. Сначала он проходит через компилятор, который знает как обращаться с синтаксисом языка программирования, потом через редактор связей, чтобы связать несколько скомпилированных файлов воедино. Получившийся исполняемый файл загружается в загрузчик, утилиту операционной системы, которая может присоединить к файлу некоторые дополнительные системные библиотеки. К этому моменту должно произойти преобразование символических адресов в физические. Преобразование адресов может происходить на этапе редактора связей либо загрузчика. Однако это означает, что программа не сможет быть никуда перемещена в оперативной памяти.

Связывание адресов

Окончательный физический адрес в таком случае присваивается блоком управления памятью во время выполнения программы.

Символьное адресное пространство -- совокупность доступных идентификаторов в определенном языке программирования.

Физическое адресное пространство -- адреса ячеек памяти, которые реально присутствуют в системе.

Логическое адресное пространство -- существует для связывания адресов на этапе выполнения. Блок управления памятью преобразовывает логические адреса в реальные адреса ячеек оперативной памяти.

Схема с фиксированными разделами

Схема памяти, которая раньше широко использовалась и предполагала собой разделение памяти на разделы определенного размера. В одном разделе можно было разместить только один процесс.

Недостатки: возникали ситуации, когда память, выделенная разным процессам, полностью ими не используется, простаивает, но при этом есть потенциальная свободная память, куда можно положить очередной процесс, но сделать этого нельзя.

Способы организации больших программ

Одной из проблемой в то время был вопрос о том, как разместить в памяти программу, которая туда не помещается.

Решение первое -- это оверлейная структура. Она предполагает, что одновременно вся программа не исполняется, а исполняется лишь ее часть, поэтому можно подгружать программу по частям по мере ее выполнения. Однако забота об этом в основном ложилась на плечи программиста.

Решение второе -- чтобы загружать только те процедуры, которые реально были вызваны в ходе исполнения программы. Это подход динамических библиотек или динамической загрузки процедур.