Менеджер памяти — часть компьютерной программы (как прикладной, так и операционной системы), обрабатывающая запросы на выделение и освобождение оперативной памяти или (для некоторых архитектур ЭВМ) запросы на включение заданной области памяти в адресное пространство процессора.
Основное назначение менеджера памяти в первом смысле — реализация динамической памяти.
Менеджеры памяти часто образуют иерархию: нижестоящие менеджеры задействуют какие-либо закономерности выделения-освобождения памяти и этим снижают нагрузку на вышестоящие. Например:

- **Системный**. Сверху находится менеджер памяти, встроенный в ОС. Он вносит ту или иную страницу в адресное пространство процесса — а значит, работает с дискретностью в 4 килобайта и очень медленный.
- **Принадлежащий процессу.** Менеджер памяти, встроенный в стандартную библиотеку языка программирования, берёт у ОС блоки памяти «оптом» и раздаёт их сообразно с нуждами программиста. При этом он знает, что память отдаётся только одному процессу — а значит, синхронизация потоков производится не мютексами, а фьютексами. И переключение в режим ядра происходит в двух случаях: либо когда «оперативного запаса» памяти не хватает и нужно обратиться к ОС, либо когда один из потоков «натыкается» на занятый фьютекс.
- **Специализированные**. Некоторые динамические структуры данных, например, std::vector, также берут память у стандартной библиотеки с запасом (например, блоками по 16 элементов). Таким образом, элементы добавляются по одному, но обращение к вышестоящему менеджеру происходит один раз за 16 элементов. Объектный пул выделяет память под объекты конкретного типа и удобен, если они выделяются-освобождаются в больших количествах, и т. д.

Страничная память — способ организации виртуальной памяти, при котором единицей отображения виртуальных адресов на физические является регион постоянного размера (т. н. *страница*). Типичный размер 4096 байт, для некоторых архитектур до 128 КБ.
Поддержка такого режима присутствует в большинстве 32битных и 64битных процессоров. Такой режим является классическим для почти всех современных ОС, в том числе Windows и семейства UNIX. Широкое использование такого режима началось с процессора VAX и ОС VMS с конца 70х годов (по некоторым сведениям, первая реализация). В семействе x86 поддержка появилась с поколения 386, оно же первое 32битное поколение.

Решаемые задачи

- поддержка изоляции процессов и защиты памяти путём создания своего собственного виртуального адресного пространства для каждого процесса
- поддержка изоляции области ядра от кода пользовательского режима
- поддержка памяти «только для чтения» и неисполняемой памяти
- поддержка отгрузки давно не используемых страниц в область подкачки на диске (см. свопинг)
- поддержка отображённых в память файлов, в том числе загрузочных модулей
- поддержка разделяемой между процессами памяти, в том числе с копированием-по-записи для экономии физических страниц
- поддержка системного вызова fork() в ОС семейства UNIX

Страничная организация (paging) – стратегия управления памятью, при которой:

- логическая память делится на **страницы** – смежные области одинаковой длины, обычно – степень 2 (например, 512 слов);
- физическая память, соответственно, делится на **фреймы** такого же размера;
- распределение логической памяти происходит с точностью до страницы;
- физическая память процесса может не быть непрерывной;

- связь между логической и физической памятью процесса осуществляется с помощью **таблицы страниц** – системной структуры, выделяемой процессу для **трансляции** его **логических адресов в физические**.

При страничной организации ОС хранит информацию обо всех свободных фреймах. Поскольку память выделяется с точностью до страницы, возможна внутренняя фрагментация (см. п. 16.5).

Цели страничной организации – обеспечить возможность не смежного распределения физической памяти для процессов, а также расширить пространство логической памяти.

При страничной организации логический адрес обрабатывается системой особым образом – как структура **(p, d)**:его старшие разряды обозначают **номер страницы**, младшие – **смещение внутри страницы. Номер страницы (p)** трактуется как индекс в таблице страниц, соответствующий элемент которой содержит базовый **адрес начала страницы в физической памяти**. **Смещение внутри страницы (d)** добавляется к ее базовому адресу. В результате формируется физический адрес, передаваемый в устройство управления памятью.