***Выделение памяти процессам: функция VirtualAlloc, куча процесса, трансляция адресов.***

***Разработайте службу, которая каждые 60 секунд выполняет считывание целого числа N из файла input.txt. При наличии числа в файле служба записывает в файл output.txt N случайных чисел в диапазоне от 0 до 25 и удаляет число N из файла input.txt.***

**Функция VirtualAlloc** используется для резервирования региона памяти в адресном пространстве процесса или её выделения. Память, выделенная этой функцией, автоматически инициализирована до нуля.

**Куча процесса (heap)** — это пул памяти какого-либо процесса. Когда программе требуется блок памяти, она вызывает функцию, выделяющую память из кучи. По умолчанию размер кучи составляет 1 МБ, но при компиляции приложения или в ходе выполнения процесса может быть изменён.

**Трансляция адресов** происходит следующим образом: при резервировании памяти выделяется непрерывный диапазон виртуальных адресов. Когда программе понадобится эта память, она снова вызовет VirtualAlloc, чтобы передать память из этого региона. Теперь начальный и конечный адреса региона округляются до значений, кратных 4 КБ, и в файле подкачки выделяются соответствующие страницы, а также создаётся нужная таблица страниц.

**VirtualAlloc: Теория низкоуровневого выделения виртуальной памяти**

Виртуальное адресное пространство: VirtualAlloc оперирует непосредственно с виртуальным адресным пространством процесса. Это абстрактное пространство адресов, которое процесс “видит” как свою собственную память, независимо от физической памяти. Оно является концептуальным и не соответствует напрямую физической оперативной памяти или адресам на жестком диске.

**Резервирование и выделение:**

Резервирование (MEM\_RESERVE): Выделяет блок виртуальных адресов в адресном пространстве процесса, но не ассоциирует его с физической памятью. Этот блок просто “занимается” и не может быть использован другими областями виртуальной памяти.

Выделение (MEM\_COMMIT): Привязывает ранее зарезервированный блок виртуальных адресов к физической памяти (либо к оперативной памяти, либо к файлу подкачки). Это делает блок доступным для использования программой.

Постраничная организация: Выделение памяти с помощью VirtualAlloc происходит в терминах страниц памяти. Страница памяти – это фиксированный блок памяти (обычно 4КБ). Зарезервированные и выделенные блоки виртуальной памяти всегда являются кратными размеру страницы.

Защита памяти (flProtect): VirtualAlloc позволяет устанавливать защиту памяти для выделенных блоков. Это определяет, какие операции (чтение, запись, выполнение) могут быть выполнены над выделенной памятью. Механизмы защиты памяти работают на уровне страниц.

Гранулярность и управление: VirtualAlloc является низкоуровневой функцией, требующей ручного управления выделением и освобождением блоков памяти. Необходимо самостоятельно отслеживать, какие диапазоны адресов были зарезервированы и выделены, и вовремя их освобождать.

**Куча процесса (Process Heap): Теория динамической памяти процесса**

Абстракция над VirtualAlloc: Куча процесса является более высокоуровневой абстракцией, построенной поверх механизмов VirtualAlloc. Она упрощает динамическое выделение памяти в процессе.

Менеджер кучи: Система предоставляет менеджер кучи, который управляет выделением и освобождением блоков памяти внутри кучи. Он поддерживает списки свободных и занятых блоков, оптимизируя использование памяти.

Области памяти: Куча, как правило, представляет собой одну или несколько областей памяти, выделенных с помощью VirtualAlloc, но управляемых менеджером кучи.

Куча по умолчанию и пользовательские кучи: Каждый процесс имеет кучу по умолчанию (получаемую через GetProcessHeap). Можно также создавать дополнительные кучи через HeapCreate.

Операции с кучей: Функции HeapAlloc, HeapFree, HeapReAlloc и другие предоставляют интерфейс для работы с кучей, включая:

Выделение: Запрашивает блок памяти из кучи.

Освобождение: Возвращает блок памяти обратно в кучу.

Изменение размера: Изменяет размер ранее выделенного блока.

Снижение фрагментации: Менеджер кучи стремится минимизировать фрагментацию памяти (непреднамеренное разделение доступной памяти на мелкие несмежные блоки), но полная дефрагментация не всегда возможна.

Управление метаданными: Менеджер кучи хранит метаданные для каждого выделенного блока, включая его размер и состояние (занят/свободен).

**Трансляция адресов: Теория отображения виртуальной памяти на физическую**

Виртуальный адрес: Адрес, который использует программа для обращения к памяти. Этот адрес является абстракцией и не соответствует физическому адресу.

Физический адрес: Адрес, который используется контроллером памяти для обращения к физической оперативной памяти или файлу подкачки.

Блок управления памятью (MMU): Аппаратный компонент CPU, отвечающий за трансляцию виртуальных адресов в физические.

Таблицы страниц: Структуры данных в оперативной памяти, которые хранят соответствие между виртуальными и физическими адресами. Каждая запись в таблице страниц соответствует странице памяти.

Иерархия таблиц страниц: Для управления большим виртуальным адресным пространством таблицы страниц могут быть многоуровневыми. Это экономит память, необходимую для хранения таблиц страниц.

**Процесс трансляции:**

Процессор генерирует виртуальный адрес.

MMU использует виртуальный адрес как индекс в таблице страниц.

Таблица страниц предоставляет соответствующий физический адрес.

MMU передает физический адрес контроллеру памяти.

Защита памяти: Таблицы страниц также хранят информацию о правах доступа (чтение, запись, выполнение) для каждой страницы. MMU использует эту информацию для контроля доступа к памяти.

Файл подкачки: В случаях нехватки физической оперативной памяти, страницы памяти могут быть перемещены в файл подкачки на жестком диске. MMU также участвует в управлении подкачкой.

TLB (Translation Lookaside Buffer): Кэш MMU для ускорения процесса трансляции адресов. Он хранит недавно использованные соответствия виртуальных и физических адресов, снижая необходимость обращаться к таблицам страниц каждый раз.