

1. Типы адресов.....	1
2. Методы распределения памяти.....	1
2.1. без использования внешней памяти.....	1
2.2. с использованием дискового пространства.....	1
2.2.1. Своппинг.....	2
2.2.2. Виртуальная память.....	2

1. Типы адресов.

Для идентификации переменных и команд используются:

- Символьные имена присваивает пользователь при написании на алгоритмическом языке или ассемблере.
- Виртуальные адреса вырабатывает транслятор, переводящий программу на машинный язык. Совокупность виртуальных адресов процесса называется виртуальным адресным пространством.
- Физические адреса соответствуют номерам ячеек оперативной памяти, где в действительности расположены или будут расположены переменные и команды. Переход от виртуальных адресов к физическим может осуществляться двумя способами:
 - Замену виртуальных адресов на физические делает специальная программа – **перемещающий загрузчик**.
 - Программа загружается в память в неизменном виде (с виртуальными адресами), а преобразование виртуального адреса в физический производится операционной системой «на лету».

2. Методы распределения памяти.

2.1. без использования внешней памяти.

-. В однозадачной ОС – зависит от программиста и задач.

А. Фиксированными разделами. Память делится на равные части фиксированного размера.

Б. Динамическими разделами. Каждой вновь поступающей задаче выделяется необходимая ей память. Если достаточный объем памяти отсутствует, то задача ждет, пока освободится достаточный непрерывный объем памяти.

В. Перемещаемыми разделами. Принцип тот же что у методов динамических разделов, только периодически (либо по таймеру либо когда фрагментация памяти больше порогового значения) происходит перемещение всех занятых разделов либо в сторону старших либо в сторону младших адресов, так чтобы вся свободная память образовывала единую свободную область. Недостаток – требуется значительное время на выполнение процедуры сжатия.

2.2. с использованием дискового пространства.

Уже достаточно давно пользователи столкнулись с проблемой размещения в памяти программ, размер которых превышал имеющуюся в наличии свободную память. Решением было разбиение программы на части, называемые *оверлеями*. 0-ой оверлей начинал выполняться первым. Когда он заканчивал свое выполнение, он вызывал другой оверлей. Все оверлеи хранились на диске и перемещались между памятью и диском средствами операционной системы. Однако разбиение программы на части и планирование их загрузки в оперативную память должен был осуществлять программист.

Существуют два основных подхода к управлению памятью, зависящие от доступного аппаратного обеспечения:

1. своппинг(swapping). Это самая простая стратегия заключается в том, что каждый процесс полностью переносится в память, работает некоторое время и затем целиком возвращается на диск.
2. виртуальная память. Преимущества данного подхода заключается в том, что позволяет работать программам даже в том случае, когда они только частично находятся в оперативной памяти.

Хороших результатов можно достигнуть комбинируя оба этих способа.

2.2.1. Своппинг

При своппинге, в отличие от страничного, сегментного и сегментно-страничного методов реализации виртуальной памяти, процесс перемещается между памятью и диском целиком, то есть в течение некоторого времени процесс может полностью отсутствовать в оперативной памяти. Существуют различные алгоритмы выбора процессов на загрузку и выгрузку, а также различные способы выделения оперативной и дисковой памяти загружаемому процессу. При его реализации могут использоваться методы без использования дискового пространства.

2.2.2. Виртуальная память

Виртуальная память - это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся оперативную память; для этого виртуальная память решает следующие задачи:

*размещает данные в запоминающих устройствах разного типа;

*перемещает по мере необходимости данные между запоминающими устройствами разного типа;

*преобразует виртуальные адреса в физические.

*Все эти действия выполняются **автоматически**, без участия программиста*, то есть механизм виртуальной памяти является прозрачным по отношению к пользователю.

Хотя были и чисто программные методы реализации виртуальной памяти это направление получило наиболее широкое развитие после получения соответствующей аппаратной поддержки.

Наиболее распространенными реализациями виртуальной памяти является страничное, сегментное и странично-сегментное распределение памяти, а также своппинг.

2.2.2.1. Страничная организация виртуальной памяти

В наиболее простом и наиболее часто используемом случае страничной виртуальной памяти каждая виртуальная память (виртуальная память каждого процесса) и физическая основная память представляются состоящими из наборов блоков или страниц одинакового размера. Для удобства реализации размер страницы всегда выбирается равным числу, являющемуся степенью 2. Тогда если общая длина виртуального адреса – N , (это тоже всегда некоторая степень 2 – 16, 32, 64), а размер страницы есть 2^M , то виртуальный адрес рассматривается как структура Ю состоящая из двух полей – первое поле занимает $(N-M+1)$ разрядов адреса и задает номер страницы виртуальной памяти, второе поле занимает $M-1$ разрядов и задает смещение внутри страницы до адресуемого элемента памяти (в большинстве случаев – байта).

2.2.2.2. Сегментная организация

При сегментной организации виртуальный адрес по-прежнему состоит из двух полей – номера сегмента и смещения внутри сегмента. Отличие от страничной организации заключается

в том, что сегменты виртуальной памяти могут быть разного размера, а также для каждого сегмента в отдельности устанавливать различные права доступа.

Сегментная организация виртуальной памяти позволяет использовать одни и те же сегменты данных и программного кода в виртуальной памяти разных задач (для каждой виртуальной памяти есть отдельная таблица сегментов, но для совместно используемых сегментов поддерживаются общие таблицы страниц). Недостатком является более медленное по сравнению со страничной организацией преобразование адреса.

2.2.2.3. Странично-сегментная организация

Странично-сегментное распределение представляет собой комбинацию обоих методов и как следствие сочетает в себе достоинства обоих подходов. При сегментно-страничной организации виртуальной памяти происходит двухуровневая трансляция виртуального адреса в физический.