

# 1 Продолжение предыдущей лекции

Очень важно понимать, что преобразование адресов выполняется на каждой команде, а то и несколько раз, особенно если в команде есть косвенная адресация.

[картинка]

Преобразование осталось, но изменило систему программирования. Появились адресно-независимые программы (позволило перемещать программы в памяти). Изменилось все.

Когда перемещать программу? Два подхода: если в памяти для загрузки не удавалось найти раздела необходимого раздела, и [...].

Увеличение размеров ОП привело к росту размеров программ. Таким образом, размещение программы единым сегментом стало затратным.

Возникла идея разделения адресного пространства процесса на участки равного размера - страницы. Все страницы программы загружались в ОП. В результате, эта идея реализовывалась (первоначально) через создание таблицы страниц, в которой указывались смещения:

[картинка]

Сама таблица страниц находится в ОП.

Появилось понятие фрейма - страница адресного пространства процесса вставляется в «рамку» сегмента.

А может ли выполняться программа, которая в памяти находится не полностью? Данный вопрос привел к появлению виртуальной памяти.

## 2 Виртуальной памяти

Существует три схемы управления виртуальной памятью: выделение памяти страницами по запросу; выделение памяти сегментами по запросу; выделение памяти сегментами, поделенными на страницы, по запросу.

Запрос возникает, когда процессор обращается к команде или данным, отсутствующим в физической памяти. Запросы реализованы в виде прерываний.

### 2.1 Страничная память

Когда процесс создается, первое, что делается, - это выделение страниц процессу. Далее процесс начинает выполняться команда за командой. Но в какой-то момент процессор обращается к данным или команде, отсутствующим в памяти - возникает страничное прерывание.

Чтобы процесс смог продолжить свое выполнение, страница должна быть загружена в память. При этом выполняется преобразование, при котором смещение берется из программы.

Для каждого процесса системой выделяется виртуальное адресное пространство.

Для того, чтобы загрузить виртуальную таблицу в физическую память, необходимо выполнять преобразования.

Для выделения памяти страницами по запросу были предложены три вида: прямое отображение, ассоциативное и ассоциативно-прямое.

1) прямое

[картинки]

2) ассоциативное

[картинки]

Ассоциативная паять - память регистровая (потому и дорогая). Устроена таким образом, что за один такт происходит сравнение с всеми

ключами и поиск нужной страницы. Ключом является смещение.

[картинка со схемой совпадения]

### 3) ассоциативно-прямое

[картинки]

Сначала страница ищется в кэше. Если не найдена, то происходит обращение к таблице страниц. При этом в кэше должны храниться адреса физические адреса, которым были недавние обращения. Должно выполняться замещение, если страница в кэше не найдено. В результате даже небольшой кэш обеспечивает до 98% эффективных обращений.

[...]

## 2.2 Алгоритмы вытеснения страниц

### 1) Выталкивание случайной страницы

Характеризуется малыми накладными расходами. Не является дискриминационным, так как любая страница может быть вытеснена, и вероятность вытеснения у каждой страницы одинаковой. Применяется редко. Может быть вытолкнута частная страница или только что загруженная.

### 2) FIFO

Для реализации каждой странице или приписывается временная метка, или организуется связный список страниц типа FIFO. Выталкивается страница, которая дольше всего находится в памяти. Может быть вытолкнута часто используемая страница.