

Раздел 1.
Введение в архитектуру
ЭВМ.

1.1. Понятие архитектуры
ЭВМ и общие механизмы
функционирования.

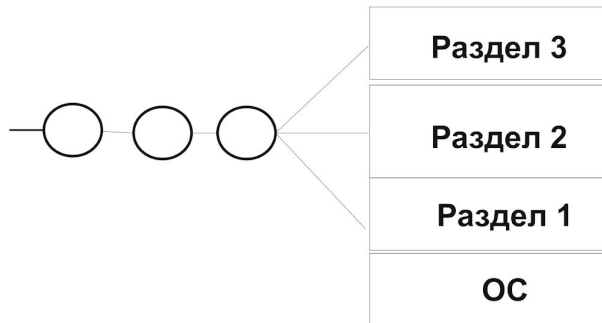
Простейшие схемы управления памятью

Простые методы управления памятью:

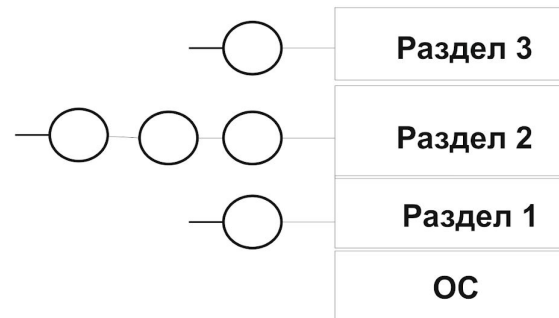
1. Каждый процесс пользователя полностью помещается в основную память, занимает непрерывную область памяти, а система принимает к обслуживанию дополнительные пользовательские процессы до тех пор, пока все они одновременно помещаются в основной памяти.
2. **«Простой свопинг»:** система размещает каждый процесс в основной памяти целиком, но иногда, на основании некоторого критерия, целиком сбрасывает образ некоторого процесса из основной памяти во внешнюю и заменяет его в основной памяти образом другого процесса. Выгруженный процесс может быть возвращен в то же самое адресное пространство или в другое. Это ограничение диктуется методом связывания. Для схемы связывания на этапе выполнения можно загрузить процесс в другое место памяти.

Схема с фиксированными разделами:

- (a) – с общей очередью процессов,
(b) – с отдельными очередями процессов



(a)



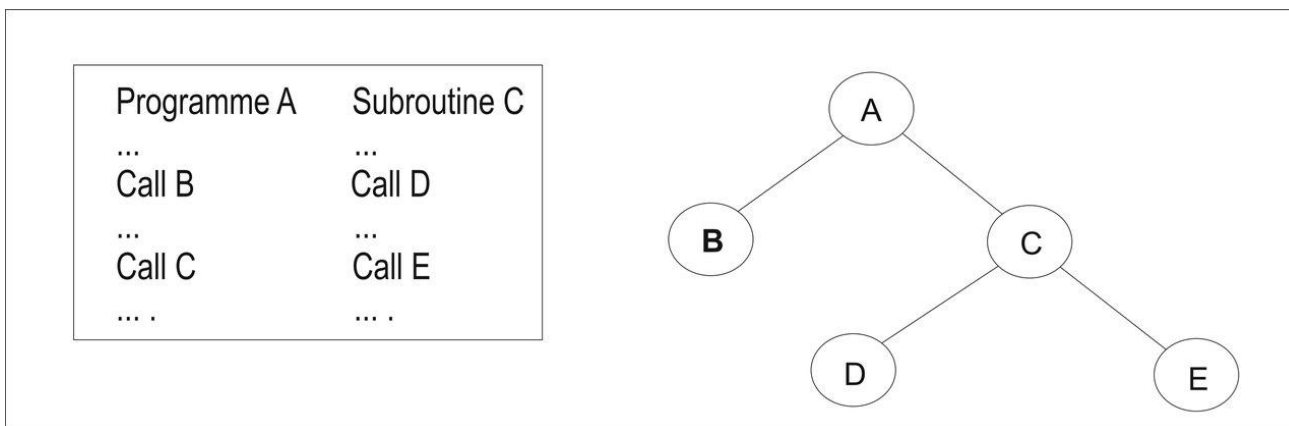
(б)

Недостатки:

1. число одновременно выполняемых процессов ограничено числом разделов;
2. данная схема сильно страдает от **внутренней фрагментации**

Оверлейная структура

Техника оверлей (overlay) или **организация структуры с перекрытием** предполагает держать в памяти только те инструкции программы, которые нужны в данный момент.



Можно поочередно загружать в память ветви:

A-B,

A-C-D

и

A-C-E программы.

Коды ветвей *оверлейной структуры* программы находятся на диске как абсолютные образы памяти и считываются драйвером оверлеев при необходимости.

Для описания *оверлейной структуры* обычно используется язык overlay description language.

Пример:

Файл с деревом вызовов внутри программы для данной схемы (файл с расширением *.odl*):

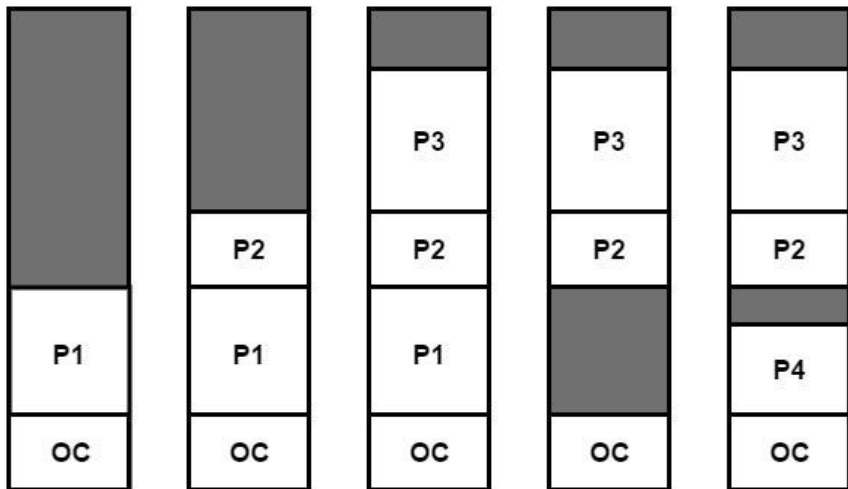
A-(B,C)

C-(D,E)

Схема с переменными разделами

Схема динамического распределения:

1. вначале вся память свободна и не разделена заранее на разделы;
2. вновь поступающей задаче выделяется строго необходимое количество памяти, не более;
3. после выгрузки процесса память временно освобождается;
4. по истечении некоторого времени память представляет собой переменное число разделов разного размера;
5. смежные свободные участки могут быть объединены.



*Динамика распределения памяти между процессами
(серым цветом показана неиспользуемая память)*

Три стратегии размещения процессов в памяти:

1. **Стратегия первого подходящего (First fit).**
2. **Стратегия наиболее подходящего (Best fit).**
3. **Стратегия наименее подходящего (Worst fit).**

Типовой цикл работы менеджера памяти:

1. анализ запроса на выделение свободного участка (раздела),
2. выбор его среди имеющихся в соответствии с одной из стратегий,
3. загрузка процесса в выбранный раздел,
4. изменения таблиц свободных и занятых областей.

Одно из решений проблемы **внешней фрагментации** – организовать сжатие, то есть перемещение всех занятых (свободных) участков в сторону возрастания (убывания) адресов, так, чтобы вся свободная память образовала непрерывную область. Этот метод иногда называют **схемой с перемещаемыми разделами**.

Страничная память

Логическое и физическое адресные пространства - это наборов блоков или *страниц* одинакового размера.



Образуются логические страницы → физические страницы (страничные кадры).

Страницы имеют фиксированную длину = степени числа 2 и не могут перекрываться.

Каждый *кадр* содержит одну *страницу* данных.

Внешняя фрагментация отсутствует, а потери из-за *внутренней фрагментации*, ограничены частью последней *страницы* процесса.

Логический адрес в страничной системе – это упорядоченная пара (p, d) , где:

p – номер *страницы* в виртуальной памяти,

d – смещение в рамках *страницы* p , на которой размещается адресуемый элемент.

Интерпретация логического адреса

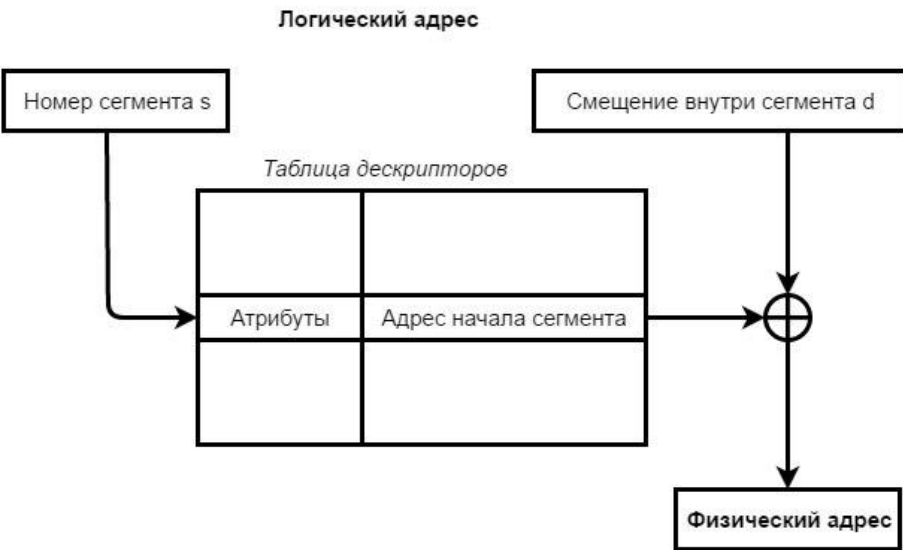
Если выполняемый процесс обращается к логическому адресу $v = (p, d)$, механизм отображения ищет номер *страницы* p в *таблице страниц* и определяет, что эта *страница* находится в страничном кадре p' , формируя реальный адрес из p' и d .

Система отображения логических адресов в физические сводится к системе отображения логических *страниц* в физические и представляет собой таблицу страниц, которая хранится в оперативной памяти.



Таблица страниц адресуется специальным регистром процессора и позволяет определить номер кадра по логическому адресу. При помощи атрибутов, записанных в строке *таблицы страниц*, можно организовать *контроль доступа к конкретной странице и ее защиту*. Для управления *физической памятью* ОС поддерживает структуру *таблицы кадров*. Она имеет одну *запись* на каждый физический кадр, показывающий его состояние.

Сегментная и сегментно-страничная организация памяти



Преобразование логического адреса при сегментной организации памяти

Логический адрес – упорядоченная пара $v = (s, d)$, где:
 s – номер сегмента,
 d – смещение внутри сегмента.

Виртуальный адрес является двумерным и состоит из двух полей:

1. номера сегмента,
2. смещения внутри сегмента.

Логическое адресное пространство – набор сегментов.

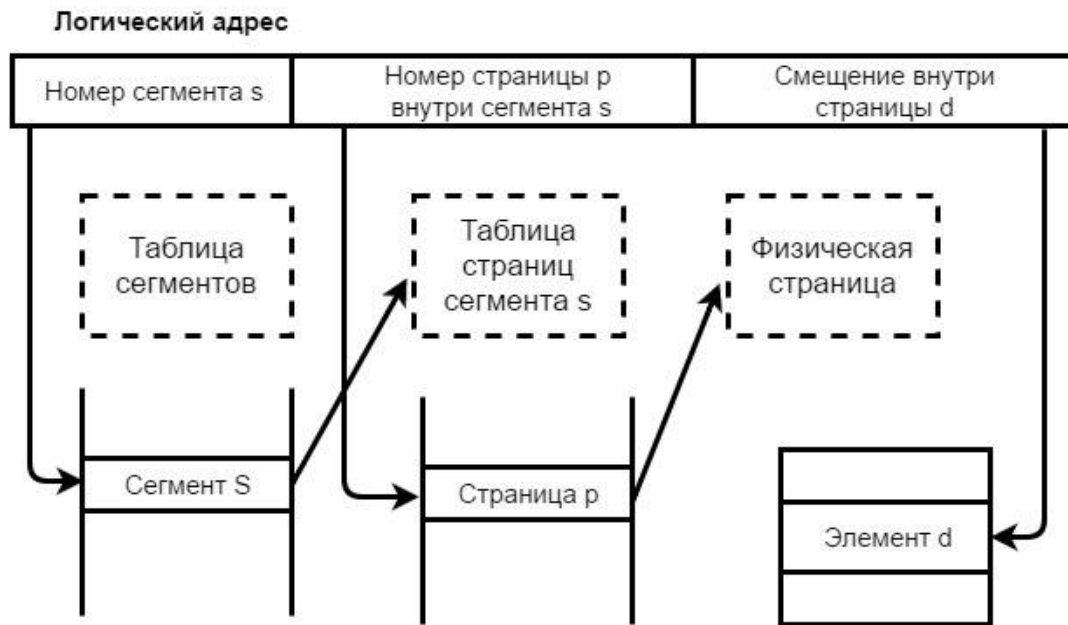
Каждый сегмент имеет:

- имя,
- размер,
- уровень привилегий,
- разрешенные виды обращений,
- флаги присутствия и т.д.

Элемент таблицы сегментов содержит:

- физический адрес начала сегмента,
- длину сегмента.

Сегментно-страничная организация памяти



Упрощенная схема формирования физического адреса при сегментно-страничной организации памяти

При сегментно-страничной организации памяти происходит **двухуровневая трансляция виртуального адреса в физический**.

Логический адрес состоит из трех полей:

- номера сегмента логической памяти,
- номера страницы внутри сегмента, смещения внутри страницы.

Поэтому используются **две таблицы отображения:**

- *таблица сегментов*, связывающая номер сегмента с таблицей страниц,
- отдельная *таблица страниц* для каждого сегмента.