*[3]Загрузчик – системная программа, выполняющая загрузку готового к исполнению модуля в физическую ОП.*

Функции загрузчика:

1. распределение ОП;

Для размещения программы в оперативной памяти должно быть найдено и выделено свободное место в памяти. Для выполнения этой функции Загрузчик обычно обращается к операционной системе, которая выполняет его запрос на выделение памяти в рамках общего механизма управления памятью.

1. перемещение программы;

**Функция перемещения** необходима потому, что программа на любом языке разрабатывается в некотором виртуальном адресном пространстве, в котором адресация ведется относительно начала программной секции. При написании программы и при ее трансляции, как правило, неизвестно, по какому адресу памяти будет размещена программа (где система найдет свободный участок памяти для ее размещения). Поэтому в большинстве случаев в командах используется именно адреса меток и данных. Однако, в некоторых случаях в программе возникает необходимость использовать реальные адреса, которые определяться только после загрузки. Все величины в программе, которые должны быть привязаны к реальным адресам, должны быть настроены с учетом адреса, по которому программа загружена.

1. связывание модулей;

**Функция связывания** состоит в компоновки программы из многих объектных модулей. Поскольку каждый из объектных модулей в составе программы был получен в результате отдельного процесса трансляции, который работает только с одним конкретным модулем, обращения к процедурам и данным, расположенным в других модулях, в объектных модулях не содержат актуальных адресов. Загрузчик же "видит" все объектные модули, входящие в состав программы, и он может вставить в обращения к внешним точкам правильные адреса.

1. загрузка

**Функция загрузки** сводится к считыванию образа программы с диска (или другого внешнего носителя) в оперативную память.

**[4][5]Виды загрузчиков**

*Абсолютный загрузчик*

Абсолютный загрузчик выполняет запись объектов программы в ОП и передачу управления на адрес начала ее исполнения. Поскольку от абсолютного загрузчика не требуется выполнения связывания и перемещения программ его работа очень простая. Все выполняется за один просмотр. В начале для того чтобы удостоверится, что программа, переданная для загрузки, корректна и что для нее достаточно места в ОП, просматривается запись-заголовок. Затем последовательно считываются записи тело программы, и содержащийся в них объектный код помещается в ОП по указанному адресу. И как только будет прочитана запись-конец, загрузчик передает управление по адресу, заданному в качестве адреса начала исполнения программы.

[6]Предназначен для загрузки загрузочного модуля в оперативную память и передачи ему управления.

*Абсолютный загрузочный модуль* – это модуль, содержащий исполняемую программу, сгенерированную на конкретный адрес загрузки в оперативной памяти.

Абсолютный загрузочный модуль содержит три типа карт:

**H<имя модуля><длина модуля><адрес загрузки>**

**Т<адрес данных><длина данных><данные>**

**Е<точка входа>**

*Алгоритм функционирования абсолютного загрузчика:*

1. Открытие файла с загрузочным модулем и чтение карты Н. Определение имени, длины и адреса загрузки модуля.
2. Запрос ОС блока памяти с указателем адреса и указателем длины. Если память не выделена, то сообщение об ошибке и переход к пункту 6
3. Чтение очередной карты.
4. Если карта E, то формирование адреса 1-й команды, равной точке входа и переход к пункту 6
5. Обработка карты T. Считывается адрес и длина данных и производится загрузка данных из карты в оперативную память по указанному адресу. Переход к п.3
6. Конец.

Абсолютные загрузчики просты в реализации, но имеют ряд особенностей:

* задачу распределения ОП выполняет программист (с помощью директивы установки начального значения ОП);
* задачу перемещения программы выполняет компилятор;
* связывание модулей - решается программистом (call 600);

*Преимущества загрузчика:*

1. Загрузчик простой, представляет программу из одного, максимум двух циклов.
2. Маленький размер загрузчика.
3. Высокая скорость

*Недостатки*

Одним из очевидных недостатков абсолютных загрузчиков является то, что требуется определить фактически адрес начала загрузки программы до ее ассемблирования.

Выполнение программы возможно только в том случае, если блок памяти, на котором сгенерирована программа свободен.

*[7]Перемещающий загрузчик*

**Перемещаемые загрузчики** обеспечивают эффективное разделение ресурсов компьютера при одновременном выполнении нескольких независимых программ, совместно использующих ОП и другие ресурсы. На ряду с простой функцией размещение программы в ОП, данный загрузчик выполняет также перемещение программ и при этом может использовать аппаратные средства.

Сегмент программ или данных может определять символ для возможной ссылки на них из других программ, а также ссылаться на символическую информацию, определяемую другими модулями. Пример: имя программы, точки входа в программу и имена областей данных.

Ссылки на внешние символы реально встречаются при вызове процедур и при использовании глобальных данных. Как внутренне определенные символы, так и внешние используемые символы должны быть явно указаны в перемещаемых объектных модулях для использования их связывающей программой. С каждым перемещаемым сегментом должна быть связана таблица определения символов и таблица использования символов.

Таблица определения символов перечисляет все символы класса, каждая запись таблицы – это пара (DName[i],DVal[i]), где DName содержит символы, а DVal – соответствующие им значения. Значением символа обычно является перемещаемый адрес А команды или данных в пределах сегмента. Когда сегмент перемещается загрузчиком, элементы сегмента с перемещаемым адресом настраиваются единообразно на место в памяти с абсолютным адресом, определяемым символом этого элемента.

Эта таблица затем становится частью глобальной таблицы символов, которая используется, чтобы отыскать адреса для внешне определенных символов.

Перемещающий загрузчик – предназначен для загрузки перемещаемого загрузочного модуля в ОП и передачи ему управления.

[8]Перемещаемый загрузочный модуль – модуль, который содержащий подпрограммы, сформированные для загрузки с нулевого адреса и дополнительную информацию для обеспечения загрузки программы с любого адреса, т.е. обеспечение перемещаемости программы.

Перемещаемый модуль содержит:

Н <имя мод.> <дл.мод.>

Т <адр.данных> <дл.дан.> <данные>

М <адр.данных> <дл.дан.> [<формула модификации> +адр.загрузки]

Е <точка входа>

Алгоритм:

1. открытие файла с перемещаемым модулем. Чтение карты Н, определение имени модуля и его длины.

2. заказ в ОС блока памяти указанной длины. Если память выделена, то её адрес присваивается адресу загрузки модуля (АЗМ), иначе (если память занята или недостаточно памяти) сообщение об ошибке и п.п. 6

3. чтение очередной карты

4. обработка карты

4.1 Если карта Е, то вычисление и установка адреса первой выполняемой команды Ап.к.=<точка входа>+АЗМ и п.п.6

4.2 Если карта Т, то определение адреса и длины данных, затем вычисление адреса данных в ОП Адоп.=<адр.данных>+АЗМ и загрузка данных из карты по вычисленному адресу Адоп

4.3 Если карта М, то определение адреса и длины данных, затем вычисление адреса данных в ОП. Выбор по вычисленному адресу значения указанной длины и увеличение на АЗМ

5. П.п. 3

6. Конец

*[9][10]Связывающий загрузчик*

Выполняют связывание и перемещение во время загрузки.

Несмотря на то, что операции связывания и загрузки выполняются загрузчиком, работа загрузчика разделяется на две части. Первая часть вырабатывает загрузочный модуль, состоящий из всех объектных сегментов, связанных и перемещаемых вместе относительно стандартного базового адреса. Другая часть – операция загрузки – загружает модуль в основную память, настраивая адреса в соответствии с распределением памяти для модуля.

Редактор связей выполняет часть работы по распределению памяти, но основную работу по распределению памяти выполняет загрузчик. Статический принцип настройки адресов выполняет статический алгоритм распределения памяти. Вся необходимая основная память для пользовательской программы и данных назначается до начала выполнения программы, а все адреса настраиваются так, чтобы отразить это назначение. Если настройка происходит во время выполнения, непосредственно предшествуя каждому обращению к памяти, то адреса настраиваются динамически. При динамической настройке адресов связывающий загрузчик должен быть построен по более сложным алгоритмам.

Входящая информация для связывающего загрузчика состоит из набора объектных программ, т.е. управляющих секций, которые должны быть связаны друг с другом. В управляющих секциях могут использоваться внешние ссылки на имена, значения которых во входном потоке еще не были определены. В этом случае требуемое связывание не может быть выполнено до тех пор, пока не будут назначены адреса для всех требующих имен, т.е. до тех пор, пока не будет прочитана требуемая управляющая секция, поэтому связывающий загрузчик обычно выполняет два просмотра входного потока точно так же, как это делает ассемблер. Т.е. во время первого просмотра назначаются адреса для всех внешних ссылок, а во время второго – выполняет фактическая перемещение, связывание и загрузка.