DHOG68

Толкование 30

Для того что бы проргамма могла быть исполненна ОС (но не оболочка)

Должна как-то определить что прикладная программа хочет оказаться в ОП и хочет что бы ей в какойто момент времени было прередано упралвение

То есть под программа тут и в дальнейшем будкм понимать совакупность команд или интрукйий которой находится в ОП и каторой помле дешефровки этих команд процессором может заставить ЭВМ выполнить строго определеннцю посдедлвоатьельно действий

Вспомним также что ОС это совокупность опппаратных и программых стредст предназначенных для управдения ЭВМ и предаставления пользоватлям услуг по управлению утсройствовами ЭВМ

Но при этом user s являются другие приклодные программы отличные от программной части ОС.

Тоесть пользователи не мы не ктото еще а именно прикладные программы.

Будем называть **исполнимым модулем** файл состоящий из машинных команд хранящиеся на внешнем устройстве, чаще всего дополнительно к машинным командам добовляется так называемый заголовок в результате чего чаще всего формируется файл с расширением exe

Если файл состоит только из машинных команд (без всякого заголовка) то говорят что жто файл с расширением .com

Упражнение 4.1

Доказать что исполнимые файлы .ехе для 64разрядных ЭВМ содержат заголовок

Доказать означает прислать к началу след лекции ворд с доказатльеством

Доказательсвом не может быть ссылка на какойто первоисточник

На прошлой лекции шла речь о memory model говорилось о том что есть memory model **flat**

Есть так же **сегментед модел** и **реал аддрес модел** при этом

Следует иметь ввиду что при испольщовании мемори модел под названием флэт

Происходит след если есть в налиции какой-то аддрес он называется линеар аддрес то это ожначает что этот дарес указывает непосредственно на какой-то место в пространстве адресов доступных ползщователю

Тоесть аддрес непостредственно указыват на некоторое место в адресном пространстве

В том случае когда речь идет о реал аддрес модел было сказано что это модель используется и лдя 64 ЭВМ с цель возможности использваония ПО созданного для 16 32 ЭВМ в том числе ля 80 86

В этом случае аддресное простарнство разбивается на сегменты при этом логический адрес формируется из 2 состовляющих – адресс который назодится в так называемом сегмент селектор

И из так называемого **смещения** при этом этот логический аддрес формируется след образом

Есть адррес начала сегмента + некоторое смещение внутри этого сегмента

Это в том случае когда речь идет о реал аддрес мод модел в качестве селектором могут быть испольщованы сегментные регистыр CS DS SS ES где сегментные регистры прелдставляют сосбой 2х байтные физические устройства

В которых храниться число 16 раз меньшее чем реальный адрес в простарнстве адрессов

Сегментед модел и мемори модел предпологает что адресное простанство доступное в программе тоже подразделяется на области но не так как при реал адресс модел

Она эти области не обязательно располагаются последовательно они могут располагатсья так что могу вообще быть отделены друг от друга но при этом немногожно менятеся картинка появляются так называемые сегменты которые являются по определенным смыслам виртуальные или логичские

Сначала адресуются эти сегменты с помощью сегментных селекторок к котором уже относятся 86 байтные регистыр которыне для 64 рисунок 3.3

Замечание

Регистры SS CS DS ES это 2х байтовые регистры и она харакрены 8086

Для пентиум но не 64 используются регистры (E)SS

Толкование 31

Структура программ получаемых в оперативной памяти в результате загрузки в ОС испольнимых модулей типа exe com существует много вариантов оформления исходных текстов на языке ассемблер

Для получения исполнимых модулей в результате транстляции и линковки

Самых общий вариант получения исполнимого модуля типа ехе является след вариант записи исходного текса

.model smol

.stack 100h

.data

.code

Где .model это деректива транслятору с яхыка ассеблер которое говорит транстялотору какие сегменты и какого размера будут сформированы при загрузке полученого исполнимого модуля в ОП .стек это директива транслятору говорит о том что будет или должен быть испольщован при загрузке использования при загрузке модуля в ОП так называемый стек

Специальная облать памяти размер стека 256 байт

.дата директива транлятору котороая говоит о том что должен быть образован сегмент данных при чем в качестве сегмент селектора будет выбран должен быть выбран сегментный регистр DS и все данные которые будут описаны далее будут адресоваться с помощью этого сегмента

Например

ZU DB ‘Зубович’ 13, 10

.code говорит транслятору что далее будут распологаттся команды после трансляции и линковки должны будут оказаться в сегменте кода адресс будет формироваться с помощь. CS

Обычно исходный текст на языке ассемблер начинается с так называемой метки которая транслятором преабразуется

Обычно самым распространеным способом формирования адресса

Mov ax, @data

Mov ds, ax

На прямую нельзя переводить Mov ds, @data ибо только ах может это сделать

После всего этого могут быть всякие действия

В конце пишем

Mov ac, 4ch

Int 21h

В конце пишем

End begin

Послле загрузки этого в ОП в ней образуется как минимум 4 области 1 область

PSP Программ сегмент префикс

Далее появится код сегмент

Дата сегмент

Стек сегмент

При этом в книжке про ассемблер показано каким образом или какие значения при этом получают соответсвующие сегментные регистры

След образом сказано что значения нахадящиеся в регистрах DS ES получают PSP

CS аддрес начала сегмента кодов при чем в регистре IP находятся адрес первой исполнимой команды инстракшен поинтер Регистра CS сегмента стека и для движения по стеку может быть испаользован регистр SP стек поинтер при оформлении исходного текста на языке ассемблер с целью получения исполнимого модуля типа com в принципе оформелние может быть подобным как и для получения исполнимого модуля типа ехе но чаще всего в файлаъ типа ком отсутствует сегмент данных

Поэтому общая структура .model tiny далее сразу записывается сегмент кодов .code

ORG 100h котрая говорит транслятору говорит о том что программ сегмент префикс должен обязательно присутствовать и состоять из 256 байт далее обычно при формировании исполднимых модулей типа ком в связи с отсутсвием сегмента данных записывается команда джамп код

Begin: jmp cod

… данные

Cod:

… команды

Mov ah, 4ch

Int 21h

End begin

Для код тоже появяться сегменты

PSP -> CS ES DS SS

Команды и данные -> IP

Стек -> SP

Уражение 4.2

Какие мемори моделс были использованы

1) при решении вами лаб 1

2) при вывод на экран мессадж бокса

Упражение 4.3

Что не так на рисунках которые были нарисованы на доске и в книге откуда они взяты