

задачи многомодульное программирование

1. ЕСЛИ вы не открывали пример многомодульного программирования откройте - можно на виртуалке всех команды в readme файлике после возвращайтесь!

2. Синтаксис и Семантика public extrn

Синтаксис: public имя , имя

Семантика: ассемблер сохраняет информацию об этих именах в объектом модуле, делая её доступной для линковщика

Синтаксис: extrn имя:тип , имя:тип

Семантика: Ассемблер использует информацию об именах и их типе для и обозначает что в модуле их нет и сообщает об этом в объектном файле, если имя присутствует то сделает имя public

3. Будет ли ошибка?

```
include console.inc
```

```
.code
```

```
myuniquestart:
```

```
    outstrln 'HelloWorld'
```

```
    exit
```

```
end myuniquestart
```

нет не будет

4. Вопрос со звёздочкой: будет ли ошибка?

```
include console.inc
```

```
.code
```

```
start:
```

```
    outstrln 'HelloWorld'
```

```
    exit
```

```
end
```

Если да то опишите как исправить (не приписывая после end start)

будет. не задана точка входа линковщик не знает где стартовать для исправления необходимо указать ключ /Entry:_start (причём _ добавится так как .console.inc есть прописан stdcall) и сделать метку public

5. ml /c /coff /Fl main.asm - стандартная компиляция object файла для нас (или через wine) поясните за каждый ключ

/c	только компиляция
/coff	требуемый тип объектного файла
/Fl	просит компилятор создать листинг (отчёт о компиляции)

6. опишите что случится с секциями .code разных модулей после линковки в 1 загружаемый файл. Секции .code сольются в одну секцию, все смещения будут подставлены

7. MyProc proto :byte,:dword,:byte - для чего существует возможность описывать прототип опишите ситуацию

proto может понадобится для удобства при импорте из другого модуля если нам известно соглашение то ассемблер сам переименует согласно ему и этим параметрам, а мы и дальше сможем использовать MyProc а не _MyProc@12 при написании кода

8. private для чего существует

по умолчанию в ассемблере все процедуры - публичные, private явно укажет что процедура может использоваться только в этом модуле

9. найдите все ошибки исправьте после линковки и запуска должен отработать next_day взяв параметры как у прототипа

```
.686; доп модуль
.XMM

.model flat, stdcall
public next_day@12
.code
; принимает 3 параметра
next_day@12:
next_day proc private
push ebp
mov ebp, esp
...
pop ebp
ret 12
next_day endp
end

.686; основной модуль
.XMM

.model flat

.data
op1 db ?
op2 dd ?
op3 db ?
.code
Start:
    next_day proto :byte,:dword,:byte
    push op3
    push op2
    push op1
    call next_day
    ...
end Start
```

необходимо исправить .model flat, stdcall

10. схематично стрелками как на лекции обрисуйте все выходы и входы сделайте вывод о линковке подпишите внешние имена в формате stdcall (указана опция: option proc:private)

```
;main.asm
public cringe, twop@4
.data
    extrn love:real4; 7F800000h))
    cringe dw ?

.code
twop@4:
twop proc
...
twop endp

onep proc public
...
onep endp
Start:
...
end Start
```

```
;module.asm
public love

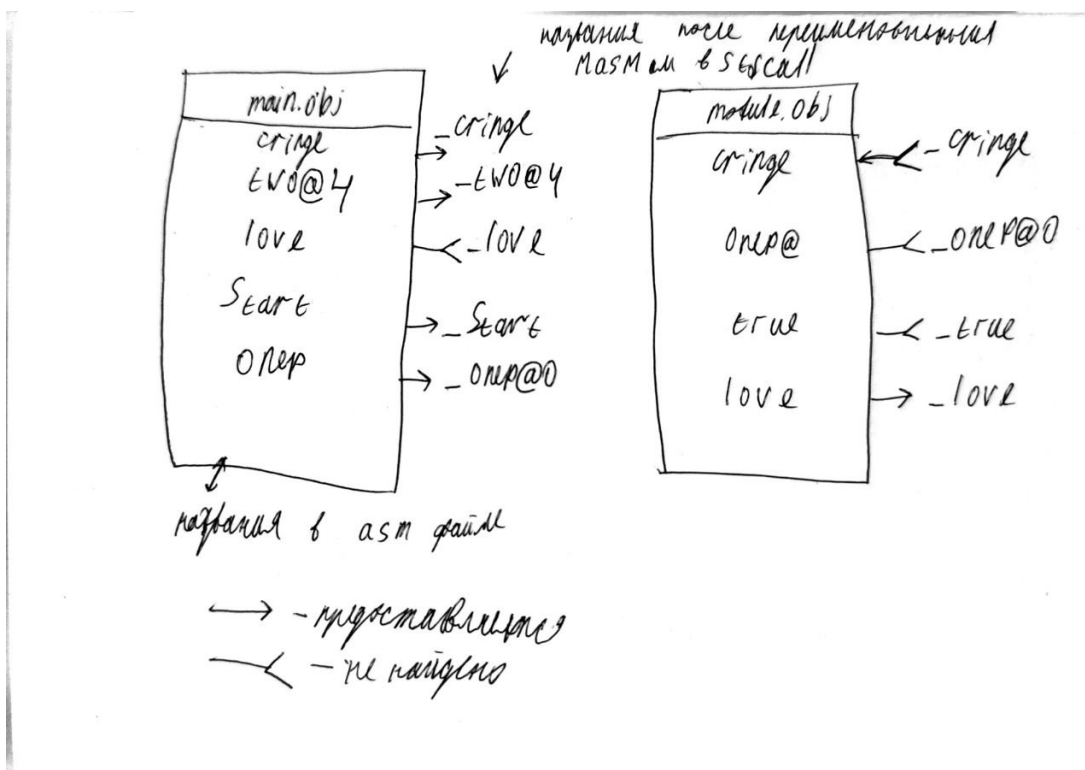
.data
    love real4 ?
    extrn cringe:word
    extrn true:byte

twop proto :dword
.code
    extrn onep@0:proc
...
end
```

пролог в обоих случаях Стандартный

.686

.model flat, stdcall
option proc:private



Проверим себя через утилиту nm

main.obj	module.obj
001220fc a @comp.id	001220fc a @comp.id
00000000 D _cringe	U _cringe
00000000 d .data	00000000 d .data
00000000 i .drectve	00000000 D _love
U _love	U _onep@0
00000002 T _onep@0	00000000 t .text
00000004 T _Start	U _true
00000000 t .text	
00000000 t _twop@0	
00000000 T _twop@4	

Если можно убрать ошибки совместимости вычеркиванием одной строки сделайте
да при линковке недостаёт одного момента 'true' вычеркнем правду(

11. дан листинг указать на месте подчёркиваний числа или буквы

00000000 .data	000000XX 88 25 ----- -	mov hapiness, ah
extrn hapiness:byte	000000XX C6 05 ----- -	mov mydata, 8
00000000 00000005 [garbage db 5 dup (?)	08	
00		
]		
00000005 2A mydata db 42;		

первый пропуски - hapiness - переменная внешняя потому будет **00000000 E** (E - extern) так как мы не знаем где будет находится hapiness
второй пропуск - mydata обратимся к листингу слева секция .data смещение от начала 00000005 так как секции будут сливаться и нам известно лишь смещение для данного объектного модуля то будет буква R итого **00000005 R**