

Компајлирање асемблерског кода

```
org 100h
mov ax, 0B800h ; komentar
mov ds, ax
mov cl, 'A'
mov ch, 1101_1111b
mov bx, 15Eh
mov [bx], cx
ret
```

После куцања кода, кликне се на дугме `compile`. Постави се питање где да се сачува компајлирани фајл. После успешне компајлације кликне се на дугме `emulate` да би се прочитао компајлирани фајл у емулатор.

Наредбе типа излазног фајла

```
#make_com#
#make_bin#
#make_boot#
#make_exe#
```

Ове наредбе се могу унети у изворни код да би се одредио захтевани излазни тип фајла. Само ако компајлер не може да одреди тип излазног фајла аутоматски и када не може да нађе ниједну од ових наредби може да затражи од програмера који тип излазног фајла треба да креира. Суштински, нема разлика између `.com` и `.bin` типова пошто су ови сирови бинарни фајлови. Излазни фајл типа `.exe` има посебна хедер на почетку фајла који се користи од стране оперативног система за одређивање неких особина егзекутабилног фајла.

Описи типова излазних фајлова

1. `#make_com#` је најстарији и најједноставнији формат егзекутабилног фајла.

Они се читавају помоћу `100h` префикса (256 бајтова). За компајлирање `.com` фајла се изабере опција **Clean** из **New** менија.

Први ред кода мора бити наредба `org 100h` и тада се аутоматски бира тип фајла при компајлирању.

Овај тип излазног фајла је подржана од стране свих Microsoft оперативних система.

2. `#make_exe#` је напреднији формат егзекутабилног фајла.

Није ограничен меморијским капацитетом и бројем сегмената.

Стек сегменти ће бити дефинисани од стране програма.

Може се изабрати `exe template` из **New** менија чиме се прави једноставан `exe` програм са предефинисаним подацима, стеком и сегментима кода.

Линија кода од које започиње извршавање фајла се дефинише од стране програмера.

Овај тип фајла се аутоматски бира ако је пронађен стек сегмент.

Овај тип излазног фајла је подржана од стране свих Microsoft оперативних система.

3. `#make_bin#` је једноставан егзекутабилан фајл.

Могу се дефинисати вредности свих регистара, сегмената и офсета за меморијске просторе где ће овај фајл бити читан.

Приликом читавања `"MY.BIN"` фајла у емулатор овај ће тражити `"MY.BINF"` фајл и учитати `"MY.BIN"` фајл у локацију која је дефинисана у `"MY.BINF"` фајлу.

Регистри су такође постављени у складу са информацијама у том фајлу.

Ако емулатор не може отворити `"MY.BINF"` фајл користе се тренутне вредности у регистрима и `"MY.BIN"` фајл се читава на тренутној `CS:IP` и стартовање започиње од вредности у `CS:IP`.

.BINF фајл ће се креирати аутоматски ако асемблер нађе било коју од следећих наредби које су на врху изворног кода за постављање вредности у регистрима:

#make_bin#	Све вредности су хексадецималне.
#LOAD_SEGMENT=1234#	Ако нису специфициране ове вредности се постављају по дифолту:
#LOAD_OFFSET=0000#	LOAD_SEGMENT = 0100
#AL=12#	LOAD_OFFSET = 0000
#AH=34#	CS = ES = SS = DS = 0100
#BH=00#	IP = 0000
#BL=00#	Ако LOAD_SEGMENT I LOAD_OFFSET нису дефинисани, онда CS и IP се користе.
#CH=00#	"#mem=..." наредбе се могу користити за писање вредности у меморију пре старта
#CL=00#	програма:
#DH=00#	#MEM=nnnn,[bytestring]-nnnn:nnnn,[bytestring]#
#DL=00#	на пример:
#DS=0000#	#MEM=1000,01ABCDEF0122-0200,1233#
#ES=0000#	све вредности су у хексадецималном облику, nnnn - за физичку адресу или
#SI=0000#	(nnnn:nnnn) за логичке адресе.
#DI=0000#	Са - се одвајају уноси, дозвољене су белине. За сваки бајт мора постојати тачно 2
#BP=0000#	карактера нпр: 0A, 12 или 00.
#CS=1234#	Ако не постоји ниједна од ових наредби у изворном коду, binf фајл се неће
#IP=0000#	направити. Када емулатор учита .bin фајл без .binf фајла он ће користити
#SS=0000#	c:\emu8086\default.binf уместо.
#SP=0000#	
#MEM=0100:FFFE,00FF-0100:FF00,F4#	

Формат типичног .BINF фајла:

8000	; load to segment.	
0000	; load to offset.	
55	; AL	Редослед линија кода је битан и треба га поштовати као у примеру.
66	; AH	Постојећи .binf фајл ће бити аутоматски пребрисан приликом поновног
77	; BL	компајлирања.
88	; BH	У случају ако је вредност за load to offset различита од 0000, мора се додати ORG
99	; CL	????h у изворном коду .BIN фајла. ???h представља читавајући офсет, који служи
AA	; CH	да компајлер правилно израчуна адресе.
BB	; DL	#make_boot# овај тип је копија првог прстена бут сектора флопи диска и ставља се
CC	; DH	наредба org 7c00h пре почетка кода.
DDEE	; DS	
ABCD	; ES	
EF12	; SI	
3456	; DI	
7890	; BP	
8000	; CS	
0000	; IP	
C123	; SS	
D123	; SP	