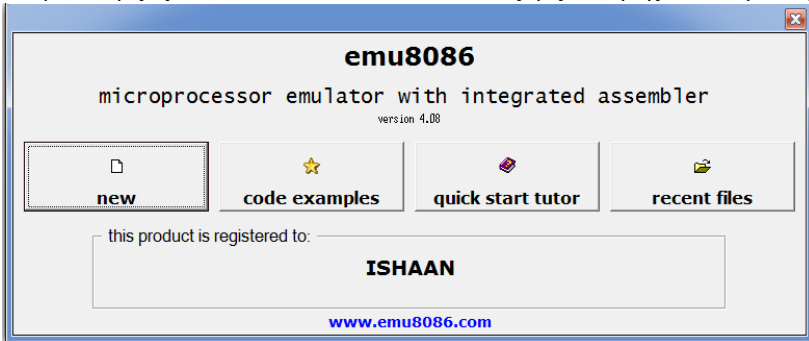
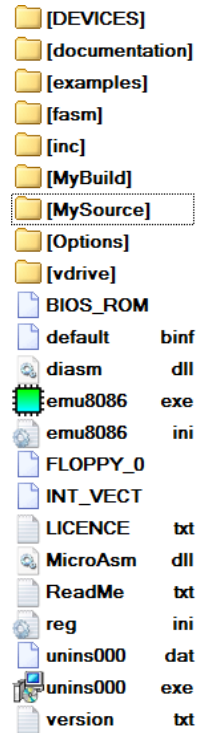


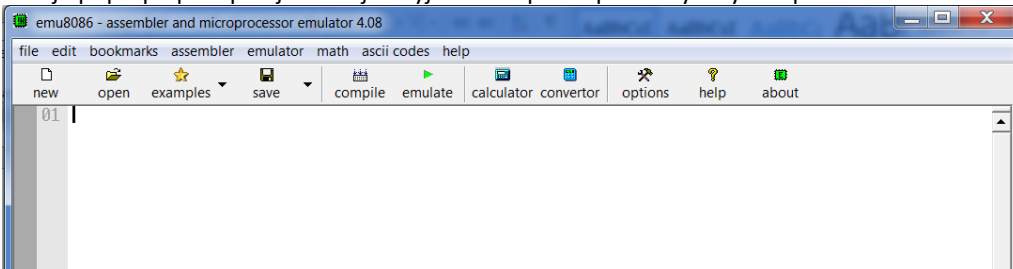
Пример рада са емулатором асемблерског језика ему8086

Пренесе се архивирани zip фајл са два фајла: текстуални и извршни фајл. Извршни фајл је setup.exe. Текстуални фајл је само информативне природе. Стартовањем setup.exe почиње инсталација која не траје дуго и чиме се инсталирају сви неопходни фајлови за рад са овим емулатором.

Интересантни су следећи директоријуми: documentation у коме се налази техничко упутство за рад са овим емулатором, MySource у коме ће се смештати сви кориснички направљени пројекти коришћењем асемблера ему8086, examples у коме се налази већина примера које ћемо користити за учење и представљање основних метода у раду са овим емулатором. Извршни фајл је ему8086.exe. Кликом на овај фајл стартује се емулатор.



Ово је поп-уп прозор који се појављује само при стартовању емулатора.

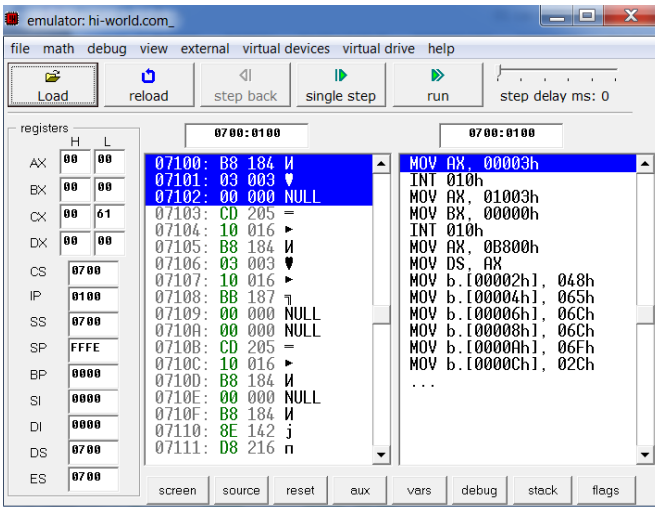


Ово је главни прозор емулатора и основни његов елемент је едитор кода који чини највећи део екрана.

Кликнем на стрелицу поред опције examples и из падајућег менија бирам Hello, world.

Појављује се исписани код траженог примера. Види се да су коментари исписани зеленом бојом и да они чине већину кода у примеру. Сам извршни код је дугачак око 100 бајтова пошто се не употребљавају прекидачи (interrupts) и пошто има само једну петљу за бојење текста. Цео преостали код се директно уписује у видео меморију.

За стартовање примера у емулатору кликне се на опцију emulate или F5. Програм ће покушати да састави (assemble) и сачува извршни фајл у c:\emu8086\MyBuild. Ако асемблер успе у креирању фајла, емулатор ће аутоматски учитати тај фајл у меморију.



Тако се отварају два прозора: први са исписаним наредбама и променама које оне изазивају у регистрима процесора и други као оригинални изворни код.

У првом се налази опција single step (F8) којом се може пролазити кроз код инструкција по инструкција и посматрати промене у регистрима. Постоји и опција step back (F6) која показује промене у корацима уназад.

Кликом на опцију Run појављује се нови екран emulator screen на којем се види излазни резултат унешеног кода.

```
original source code
28 ; f      1111      white
29
30
31
32 org 100h
33
34 ; set video mode
35 mov ax, 3          ; text mode 80x25, 16 colors, 8 pages (at
36 int 10h          ; do it!
37
38 ; cancel blinking and enable all 16 colors:
39 mov ax, 1003h
40 mov bx, 0
41 int 10h
42
43
44 ; set segment register:
45 mov ax, 0b800h
46 mov ds, ax
47
```

Процесор може схватити садржај меморије или као инструкције или као податке, не постоји разлика у појединачним бајтовима у меморији само начин на који су распоређени. Из тог разлога, могуће је да програми сами препишу по својим инструкцијама, па онда да изврше инструкције које су променили.