

# hello.asm

```

;; hello world program

.ORIG x3000

LEA R0, message      ; store address of start of message in R0
PUTS                  ; Invokes BIOS routine to output string
HALT                  ; terminates program execution

;; end of code

;; data block
message .STRINGZ "Hello World!!\n"
                ; store 'H' in address labelled "message"
                ; and then one character per memory address

.END                ; tells assembler to stop reading & assembling

```

## Simpl Window Explained:

**Simpli LC-3 Simulator Version 2.0.3**

File State View Format

Track: pc -> Addr: x3010 <no label> -> Value: x000A 10

Address	Op-code (Hex/Dec/Bin)	Label	Data
3000:	E002 -8190		R0 <- message
3001:	F022 -4062		->disp R0 zstring
3002:	F025 -4059		halt
3003:	0048 72	message	no-op (.fill 72 'H')
3004:	0065 101		no-op (.fill 101 'e')
3005:	006C 108		no-op (.fill 108 'l')
3006:	006C 108		no-op (.fill 108 'l')
3007:	006F 111		no-op (.fill 111 'o')
3008:	0020 32		no-op (.fill 32)
3009:	0057 87		no-op (.fill 87 'W')
300A:	006F 111		no-op (.fill 111 'o')
300B:	0072 114		no-op (.fill 114 'r')
300C:	006C 108		no-op (.fill 108 'l')
300D:	0064 100		no-op (.fill 100 'd')
300E:	0021 33		no-op (.fill 33 '!')
300F:	0021 33		no-op (.fill 33 '!')
3010:	000A 10		no-op (.fill 10)
3011:	0000 0		no-op (.fill 0)
3012:	0000 0		no-op (.fill 0)
3013:	0000 0		no-op (.fill 0)

**Register Table**  
Shows values stored at each register

Step	Next Line	Run	Finish Routine
R0 x3003 message	R1 x7FFF 32767	R2 x0000 0	R3 x0000 0
R4 x0000 0	R5 x0000 0	R6 x0000 0	R7 x0490 1168

CC x0200 p PC x3001 12289 Log Exec: 1

Notice:

- every line in .asm file (except pseudo-ops) is stored in memory like above.
- Pseudo-ops are run before the memory table above is filled.
  - See what .STRINGS operator have done.
- Label 'message' refers to the address x3003

## **Additional Exercises:**

### **Exercise 0.1: Step**

- After you run the example code, close simpl window and reopen it by typing: 'simpl hello.asm' again on terminal. (Alternatively, you can press Ctrl+L without closing the window to reload last file)
- Now, instead of clicking on 'Run', click on 'Step' button, this will process only a single line. Now, check the 'new' value of R0 on register table. You can 'Run' again to run the rest of the program.

### **Exercise 0.2: LEA vs LD**

To see the difference between LEA and LD.

- Change LEA to LD in the code and delete PUTS line
- rerun simpl and click 'Step' to see the 'new' value of R0 on Registry Table.
  - what value does it store? is it different from before?
  - Where does that value come from (check the data column and find the value)?