

(<https://github.com/Schweigi/assembler-simulator>)

## Code (Instruction Set (./instruction-set.html))

Fork me on GitHub

```
; Simple example
; Writes Hello World to the output

        JMP start
hello: DB "ERŠ 3CE in 4CE!" ; Variable
        DB 0          ; String terminator

start:
        MOV C, hello    ; Point to var
        MOV D, 232      ; Point to output
        CALL print
        HLT             ; Stop execution

print:                                     ; print(C:*from, D:*to)
        PUSH A
        PUSH B
        MOV B, 0

.loop:
        MOV A, [C]      ; Get char from var
        MOV [D], A      ; Write to output
        INC C
        INC D
        CMP B, [C]      ; Check if end
        JNZ .loop       ; jump if not

        POP B
        POP A
        RET
```

Assemble

## Output

## CPU &amp; Memory

## Registers / Flags

A	B	C	D	IP	SP	Z	C	F
00	00	00	00	00	E7	FALSE	FALSE	FALSE

## RAM

1F	12	45	52	160	20	33	43	45	20	69	6E	20	34	43	45
21	00	06	02	02	06	03	E8	38	1B	00	32	00	32	01	06
01	00	03	00	02	05	03	00	12	02	12	03	15	01	02	27
22	36	01	36	00	39	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Clock speed:  Instructions: Hide View: Decimal

Register addressing: A: Show B: Show C: Show D: Show

## Labels

Name	Address	Value
.loop	22	03
hello	02	45 ('E')
print	1B	32 ('2')
start	12	06