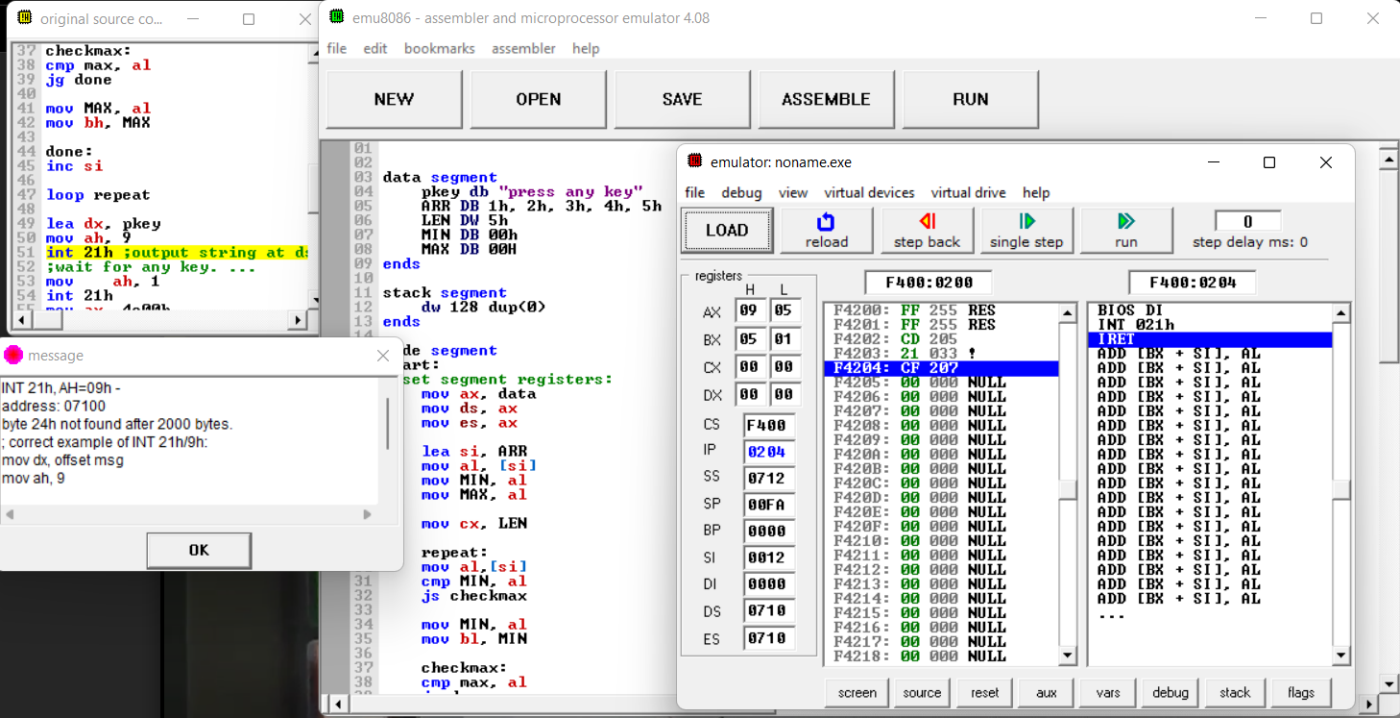
Learning

There are a number of different forms of learning as applied to artificial intelligence. The simplest is learning by trial and error. For example, a simple computer program for solving mate-in-one chess problems might try moves at random until mate is found. The program might then store the solution with the position so that the next time the computer encountered the same position it would recall the solution. This simple memorizing of individual items and procedures—known as rote learning—is relatively easy to implement on a computer. More challenging is the problem of implementing what is called generalization. Generalization involves applying past experience to analogous new situations. For example, a program that learns the past tense of regular English verbs by rote will not be able to produce the past tense of a word such as *jump* unless it previously had been presented with *jumped*, whereas a program that is able to generalize can learn the "add *ed*" rule and so form the past tense of *jump* based on experience with similar verbs.

Reasoning

To reason is to draw inferences appropriate to the situation. Inferences are classified as either deductive or inductive. An example of the former is, "Fred must be in either the museum or the café. He is not in the café; therefore he is in the museum," and of the latter, "Previous accidents of this sort were caused by instrument failure; therefore this accident was caused by instrument failure." The most significant difference between these forms of reasoning is that in the deductive case the truth of the premises guarantees the truth of the conclusion, whereas in the inductive case the truth of the premise lends support to the conclusion without giving absolute assurance. Inductive reasoning is common in science, where data are collected and tentative models are developed to describe and predict future behaviour—until the appearance of anomalous data forces the model to be revised. Deductive reasoning is common in mathematics and logic, where elaborate structures of irrefutable theorems are built up from a small set of basic axioms and rules.


original source co...
checkmax:
cmp max, al
jg done
mov MAX, al
mov bh, MAX
done:
inc si
loop repeat
lea dx, pkey
mov ah, 9
int 21h ;output string at ds
;wait for any key. ...
mov ah, 1
int 21h
message
INT 21h, AH=09h -
address: 07100
byte 24h not found after 2000 bytes.
; correct example of INT 21h/9h:
mov dx, offset msg
mov ah, 9
OK
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler help
NEW
OPEN
SAVE
ASSEMBLE
RUN
data segment
pkey db "press any key"
ARR DB 1h, 2h, 3h, 4h, 5h
LEN DW 5h
MIN DB 00h
MAX DB 00H
ends
stack segment
dw 128 dup(0)
ends
de segment
art:
set segment registers:
mov ax, data
mov ds, ax
mov es, ax
lea si, ARR
mov al, [si]
mov MIN, al
mov MAX, al
mov cx, LEN
repeat:
mov al,[si]
cmp MIN, al
js checkmax
mov MIN, al
mov bl, MIN
checkmax:
cmp max, al
emulator: noname.exe
file debug view virtual devices virtual drive help
LOAD
reload
step back
single step
run
0
step delay ms: 0
registers
H L
AX 09 05
BX 05 01
CX 00 00
DX 00 00
CS F400
IP 0204
SS 0712
SP 00FA
BP 0000
SI 0012
DI 0000
DS 0710
ES 0710
F400:0200
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205
F4203: 21 033 !
F4204: CF 207
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F4216: 00 000 NULL
F4217: 00 000 NULL
F4218: 00 000 NULL
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
screen
source
reset
aux
vars
debug
stack
flags