

Commands SO FAR

MOV EX: MOV AL, 41h → Move to the AL register 65(10)

JUMP commands:

JMP → unconditional jump

JZ → Jump if the last commands resulted in Zero

JNZ → Jump if the last commands did not result in Zero

JC → Jump if the last commands resulted in a Carry

JNC → Jump if the last commands did not result in a carry

JS → Jump if the last commands resulted in negative

JNS → Jump if the last commands resulted in positive or Zero

CMP EX: CMP AL, 45 → Compare the value in DL to 45

L: → location L in the program

; For comments.

INC EX: INC DL → increment DL by one.

DEC EX: DEC DL → Decrement DL by 1

ADD DL, 20h → $DL = (DL) + 32(10)$

SUB DL, 20H → $DL = (DL) - 32(10)$

MOV ax, 4c00h ; ends the program AX = AI + AH

INT 21h ; library call

MOV DL, Value → to display the value of DL (In ASCII)

MOV AH, 6 ; display a single character commands

INT21h

MOV AH, 1 → enter a single key from the keyboard

INT 21h → The key lands into the AL register.

Call name → Calling a procedure (Function)

RET → return Command.

MOV DX,offset msg1 ; Display a string

MOV, ah, 9

INT 21h

MUL Register EX: MUL EDX → EAX * EDX → EAX

MUL Variable EX: MUL X → EAX * X → EAX

DIV → Divides always the A register.

DIV EDX → EAX /ECX → EAX register has the result

The remainder → EDX register

FILO → First IN Last OUT

PUSH DX → Puts the content of DX into the stock

POP DX → retrieves the content from the stock into DX register.