Final Project Microprocessors

Merna Nader Abdel Malak 7608

Group 1 section 1

Project 8:

ATM MACHINE CHECKER:

The user enters in decimal his card number (16 bits) which means from 0 to 65535, and his password (4 bits), from 0 to 15.

If the data of the user matches with one of the 20 customers in the database --> output 1 ELSE 0

DATABASE:

Card number	Password
5566	0
5577	1
1234	2
1357	3
7798	4
8820	5
9934	6
7744	7
5621	8
6644	9
1389	10

1534	11
4378	12
7755	13
6699	14
4469	15
3468	1
8811	2
4334	3
2398	4

As required the card number is 16 bits, which means within the range from 0 to 65535 in decimal, and 0000H to FFFH in hexadecimal. The password is 4 bits, from 0 to 15 is decimal, 0H to FH in hexadecimal.

Code:

The code is divided into 3 parts

1.Construction of the database:

To store the 20 customers in memory , I use two arrays(table) one for card numbers and the other for passwords.

```
CARD dw 5566,5577,1234,1357,7798,8820,9934,7744,5621,6644,1389,1534,4378,7755,6699,4469,3468,8811,4334,2398
PASS db 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1,2,3,4
```

2.Reading the input:

The user enters the card number and the password. The input is in decimals, and we use procedure called SCAN_NUM is a macro copied from emu8086.inc to perform this task. For each character entered by the user, it is checked whether it is a digit from 0 to 9, then it is converted from ASCII to hexadecimal. Then, if allowed, the digit entered is shifted one place to the left by multiplying by 10 and the last result (for the first time = 0) is add to it. For each next digit the same process occurs until the user press enter button.

Then we check if the password is not 4 bits (>15) so the user will enter the password again.

```
;to check if password is out of range
cmp num2,15
ja  wrong
```

```
wrong:
      lea dx, msg8  ;msg8"password out of range!,please enter again"
      mov ah, 09h
      int 21h
      jmp start3
```

After reading both inputs from the user which are stored in memory locations labeled by num1 and num2

```
start2:

lea dx,msg1     ;msg1"Enter card number:"
mov ah, 09h
int 21h

call scan_num

; store card number:
mov num1, cx


start3:
; new line:
putc 0Dh
putc 0Ah


lea dx, msg2  ;msg2"Enter the password:"
mov ah, 09h
int 21h


call scan_num


; store password:
mov num2, cl
```

3. Validation of input:

First we defined new memory location labeled by count to use it as index to check password if we found card number for example if the user enter card number =1234 ,1234 is the third number in the Card table (array) so in this case count will be 2 as we start from 0

Second we set CX=20 as we have 20 customer to check

Finally, we set BX=0 to use it in the loop after checking each card number we increment it by 2 to check the following card number.

If the card number matches, go check for the password by setting count, if the password also matches then the customer is found, print " 1(ALLOWED)", else customer not found ,print "0(DENIED)".. If the card number is incorrect go check for the next customer by incrementing BX by 2. After checking the whole table, if the customer is not found print "0(DENIED)".

```
mov cx,20
mov bx,0
mov count,0

check:
     mov dx,num1
     cmp dx,CARD[bx]
     je  check2
     inc bx
     inc bx
     inc count
     loop check
     jmp  incorrect1
check2:
     mov dl,num2
     mov bx,count
     cmp num2,PASS[bx]
     je  allowed
     jmp incorrect2

allowed:
       lea dx, msg3
       mov ah, 09h
       int 21h
       jmp finish

incorrect1:
       lea dx,msg5
       mov ah, 09h
       int 21h

       jmp denied

incorrect2:
       lea dx, msg6
       mov ah, 09h
       int 21h
       jmp denied


denied:
       lea dx, msg4
       mov ah, 09h
       int 21h
```

Finally, the user is asked whether to end the program or to check another customer.

```
finish:

      ; new line:
      putc 0Dh
      putc 0Ah

      lea dx,msg7
      mov ah,09h
      int 21h
      call SCAN_NUM
      cmp cx,1
      jne start2
      je  exit

exit:
      mov ah,4ch
      int 21h
```

Sample Run:

```
Enter card number: 8811

Enter the password: 2

1 (ALLOWED)

press 1 to exit or other key to check another card:5
Enter card number: 5325

Enter the password: 4

Incorrect Card Number
0 (DENIED)

press 1 to exit or other key to check another card:0
Enter card number: 6699

Enter the password: 10

Incorrect Password
0 (DENIED)

press 1 to exit or other key to check another card:1
```

Code:

```
org 000
jmp start1
msg0 db "Welcome to ATM",0Dh,0Ah,'$'
msg1 db 0Dh,0Ah,"Enter card number: $"
msg2 db 0Dh,0Ah,"Enter the password: $"
msg3 db 0Dh,0Ah,"1 (ALLOWED)$"
msg4 db 0Dh,0Ah,"0 (DENIED)$"
msg5 db 0Dh,0Ah,"Incorrect Card Number$ "
msg6 db 0Dh,0Ah,"Incorrect Password$"
msg7 db 0Dh,0Ah,"press 1 to exit or other key to check another card:$"
msg8 db 0Dh,0Ah,"password out of range!,please enter again$"
num1 dw ?
num2 db ?
count dw 0
CARD dw
5566,5577,1234,1357,7798,8820,9934,7744,5621,6644,1389,1534,4378,7755,669
9,4469,3468,8811,4334,2398
PASS db 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1,2,3,4

start1:
mov dx,offset msg0
mov ah, 9
int 21h
```

```
start2:

lea dx,msg1    ;msg1"Enter card number:"
mov ah, 09h
int 21h

call scan_num

; store card number:
mov num1, cx



start3:
; new line:
putc 0Dh
putc 0Ah


lea dx, msg2  ;msg2"Enter the password:"
mov ah, 09h
int 21h
call scan_num

; store password:
mov num2, cl
```

```
;to check if password is out of range
cmp num2,15
ja  wrong
; new line:
putc 0Dh
putc 0Ah


mov cx,20
mov bx,0
mov count,0

check:
   mov dx,num1
   cmp dx,CARD[bx]
   je  check2
   inc bx
   inc bx
   inc count
   loop check
   jmp  incorrect1
check2:
   mov dl,num2
   mov bx,count
```

```
    cmp dl,PASS[bx]
    je  allowed
    jmp incorrect2

allowed:
      lea dx, msg3
      mov ah, 09h
      int 21h
      jmp finish

incorrect1:
      lea dx,msg5
      mov ah, 09h
      int 21h
       jmp denied

incorrect2:
      lea dx, msg6
      mov ah, 09h
      int 21h
      jmp denied

denied:
      lea dx, msg4
      mov ah, 09h
```

```
    int 21h
    jmp finish

wrong:
    lea dx, msg8        ;msg8"password out of range!,please enter again"
    mov ah, 09h
    int 21h
    jmp start3

finish:
   ; new line:
   putc 0Dh
   putc 0Ah
   lea dx,msg7
   mov ah,09h
   int 21h
   call SCAN_NUM
   cmp cx,1
   jne start2
   je  exit
exit:
   mov ah,4ch
   int 21h
```