



الاسم: أحمد مجدى على حسين خليفة

الرقم الجامعى: 18010189

الاسم: محمد إبراهيم محمد السعيد إبراهيم

الرقم الجامعى: 18011342

الاسم: زياد السيد رمضان

الرقم الجامعى: 18010716

الاسم: عبدالرحمن نبيل عبده محمد

الرقم الجامعى: 18010991

الاسم: محمد إبراهيم جاد سعد

الرقم الجامعى: 18011334

Microprocessor Software Project

Project #8: First-In-First-Out Queueing:





Introduction:

This program is taking an input (conditional input) to decide if someone entered the bank (input =1) so it will give him a number of the next available seat which he should set on or if someone left the bank (input =0) so it will detect that there is now an empty seat to count the available seat and organize the operation of seating in the bank.

Code:

```
#make_bin#
```

```
; BIN is plain binary format similar to .com format, but not limited to 1 segment;
```

```
; All values between # are directives, these values are saved into a separate .binf file.
```

```
; Before loading .bin file emulator reads .binf file with the same file name.
```

```
; All directives are optional, if you don't need them, delete them.
```

```
; set loading address, .bin file will be loaded to this address:
```

```
#LOAD_SEGMENT=0500h#
```

```
#LOAD_OFFSET=0000h#
```

```
; set entry point:
```

```
#CS=0500h#    ; same as loading segment
```

```
#IP=0000h#    ; same as loading offset
```

```
; set segment registers
```

```
#DS=0500h#    ; same as loading segment
```

```
#ES=0500h#    ; same as loading segment
```



```
; set stack

#SS=0500h#    ; same as loading segment
#SP=FFFEh#    ; set to top of loading segment

; set general registers (optional)

#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#

include 'emu8086.inc'

MOV BX,0000h
MOV SI,0000h
MOV DI,0000h
MOV CX,10

EMPTY:
MOV [BX],0
INC BX
LOOP EMPTY

MOV BX,0000

PRINTN "ENTER 1 IF SOMEONE ENTERED THE BANK AND 0 IF SOMEONE LEFT "
PRINTN "#####"

INTRO:

PRINTN ""
```



PRINT "THE CONDITIONAL NUMBER (0/1) IS: "

CALL SCAN_NUM

CMP CX,1

JE CHKINN

CMP CX,0

JE CHKOUT

JMP ERROR

OUTT:

MOV BX,0000

MOV [DI],0

INC DI

CMP DI,10

JE RETURNDI

JMP DETECT

ENTERED:

MOV BX,0000

MOV [SI],1

INC SI

CMP SI,10

JE RETURN SI

JMP DETECT

RETURN SI:



```
MOV SI,0
JMP DETECT
RETURNDI:
MOV DI,0
JMP DETECT
CHKINN:
MOV CX,10
CHECKIN:
CMP [BX],0
JE ENTERED
INC BX
LOOP CHECKIN
PRINTN ""
PRINTN "THERE IS NO EMPTY SEATS"
MOV BX,0
MOV SI,DI
JMP INTRO
CHKOUT:
MOV CX,10
CHECKOUT:
CMP [BX],1
JE OUTT
INC BX
LOOP CHECKOUT
PRINTN ""
PRINTN "NO ONE HERE TO LEAVE"
MOV DI,SI
```



JMP INTRO

ERROR:

PRINTN ""

PRINTN "WRONG CHOICE"

DETECT:

MOV AX,SI

ADD AX,1

PRINTN ""

PRINT "THE NEXT AVAILABLE SEAT NUMBER (1:10) IS: "

CALL PRINT_NUM

CALL INTRO

DEFINE_SCAN_NUM

DEFINE_PRINT_STRING

DEFINE_PRINT_NUM

DEFINE_PRINT_NUM_UN\$

DEFINE_PTHIS

HLT ; halt!



Simulation:

- Random experiment and if we tried to enter a wrong input.

The screenshot shows the emulator interface with the following components:

- Registers:** F400:01C0 and F400:01C1. F400:01C0 contains values for AX (00 07), BX (00 00), CX (00 00), DX (00 00), CS (F400), IP (01C0), SS (0500), SP (FFDE), BP (0000), SI (0006), DI (0002), DS (0500), and ES (0500).
- Source:** Assembly code for a checkout process, including instructions like `CMP [BX], 1`, `JE OUT1`, `INC BX`, `LOOP CHECKOUT`, `PRINTN ""`, and `PRINTN "NO ONE HERE TO`.
- Screen:** Emulator screen (80x25 chars) showing a message: "ENTER 1 IF SOMEONE ENTERED THE BANK AND 0 IF SOMEONE LEFT". Below the message is a list of seat numbers: "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 2", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 3", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 4", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 5", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 6", "THE CONDITIONAL NUMBER IS: 0", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 6", "THE CONDITIONAL NUMBER IS: 0", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 6", "THE CONDITIONAL NUMBER IS: 8", "WRONG CHOICE", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 6", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 7", "THE CONDITIONAL NUMBER IS: 1".

- If we entered 10 people and the seats are full.

The screenshot shows the emulator interface with the following components:

- Registers:** F400:01C0 and F400:01C1. F400:01C0 contains values for AX (00 01), BX (00 00), CX (00 00), DX (00 00), CS (F400), IP (01C0), SS (0500), SP (FFDC), BP (0000), SI (0000), DI (0000), DS (0500), and ES (0500).
- Source:** Assembly code for a checkout process, including instructions like `CMP [BX], 1`, `JE OUT1`, `INC BX`, `LOOP CHECKOUT`, `PRINTN ""`, and `PRINTN "NO ONE HERE TO`.
- Screen:** Emulator screen (80x25 chars) showing a message: "ENTER 1 IF SOMEONE ENTERED THE BANK AND 0 IF SOMEONE LEFT". Below the message is a list of seat numbers: "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 2", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 3", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 4", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 5", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 6", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 7", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 8", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 9", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 10", "THE CONDITIONAL NUMBER IS: 1", "THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 1", "THE CONDITIONAL NUMBER IS: 1", "THERE IS NO EMPTY SEATS", "THE CONDITIONAL NUMBER IS: 1".



- If there is no one and we tried to get someone to leave.

The screenshot shows an x86 emulator window titled "emulator: PROJECT.bin_". The registers window shows the following values:

Register	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	F400	01C0
IP	01C0	
SS	0500	
SP	FF00	
BP	0000	
SI	0000	
DI	0000	
DS	0500	
ES	0500	

The memory window shows the following code:

```

F400:01C0  F41C0: FF 255 RES
F400:01C1  F41C1: FF 255 RES
F41C2: CD 205 =
F41C3: 16 022 =
F41C4: CF 207 =
F41C5: 00 000 NULL
F41C6: 00 000 NULL
F41C7: 00 000 NULL
F41C8: 00 000 NULL
F41C9: 00 000 NULL
F41CA: 00 000 NULL
F41CB: 00 000 NULL
F41CC: 00 000 NULL
F41CD: 00 000 NULL
F41CE: 00 000 NULL
F41CF: 00 000 NULL
F41D0: FF 255 RES
F41D1: FF 255 RES
F41D2: CD 205 =
F41D3: 11 017 =
F41D4: CF 207 =
  
```

The BIOS window shows the following code:

```

BIOS DI
INT 016h
I RET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
DEC BP
ADC DI, CX
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
DEC BP
ADC AX, 000CFh
ADD [BX + SI], AL
  
```

The original source code window shows the following code:

```

107 CHECKOUT:
108 CMP [BX],1
109 JE OUT
110 INC BX
111 LOOP CHECKOUT
112 PRINTN "
113 PRINTN "NO ONE HERE TO
  
```

The Random Access Memory window shows the following data:

```

0500:0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0500:0010  E2 FA BB 00 00 50 56 EB 3D 45 4E 54 45 52 20 31
0500:0020  20 49 46 20 53 4F 4D 45 4F 4E 45 20 45 4E 54 45
0500:0030  52 45 44 20 54 48 45 20 42 41 4E 4B 20 41 4E 44
0500:0040  20 30 20 49 46 20 53 4F 4D 45 4F 4E 45 20 4C 45
0500:0050  46 54 20 00 00 BE 19 00 2E 8A 04 3C 00 74 07
0500:0060  46 B4 0E CD 10 EB F2 5E 58 50 56 EB 3C 23 23 23
  
```

The emulator screen (80x25 chars) shows the following text:

```

ENTER 1 IF SOMEONE ENTERED THE BANK AND 0 IF SOMEONE LEFT
*****
THE CONDITIONAL NUMBER IS: 0
NO ONE HERE TO LEAVE
THE CONDITIONAL NUMBER IS: 0
  
```

- Random experiment.

The screenshot shows an x86 emulator window titled "emulator: PROJECT.bin_". The registers window shows the following values:

Register	H	L
AX	00	05
BX	00	00
CX	00	00
DX	00	00
CS	F400	01C0
IP	01C0	
SS	0500	
SP	FFDC	
BP	0000	
SI	0004	
DI	0004	
DS	0500	
ES	0500	

The memory window shows the following code:

```

F400:01C0  F41C0: FF 255 RES
F400:01C1  F41C1: FF 255 RES
F41C2: CD 205 =
F41C3: 16 022 =
F41C4: CF 207 =
F41C5: 00 000 NULL
F41C6: 00 000 NULL
F41C7: 00 000 NULL
F41C8: 00 000 NULL
F41C9: 00 000 NULL
F41CA: 00 000 NULL
F41CB: 00 000 NULL
F41CC: 00 000 NULL
F41CD: 00 000 NULL
F41CE: 00 000 NULL
F41CF: 00 000 NULL
F41D0: FF 255 RES
F41D1: FF 255 RES
F41D2: CD 205 =
F41D3: 11 017 =
F41D4: CF 207 =
  
```

The BIOS window shows the following code:

```

BIOS DI
INT 016h
I RET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
DEC BP
ADC DI, CX
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
DEC BP
ADC AX, 000CFh
ADD [BX + SI], AL
  
```

The original source code window shows the following code:

```

104 CHKOUT:
105 MOV CX,10
106 CHECKOUT:
107 CMP [BX],1
108 JE OUT
109 INC BX
  
```

The Random Access Memory window shows the following data:

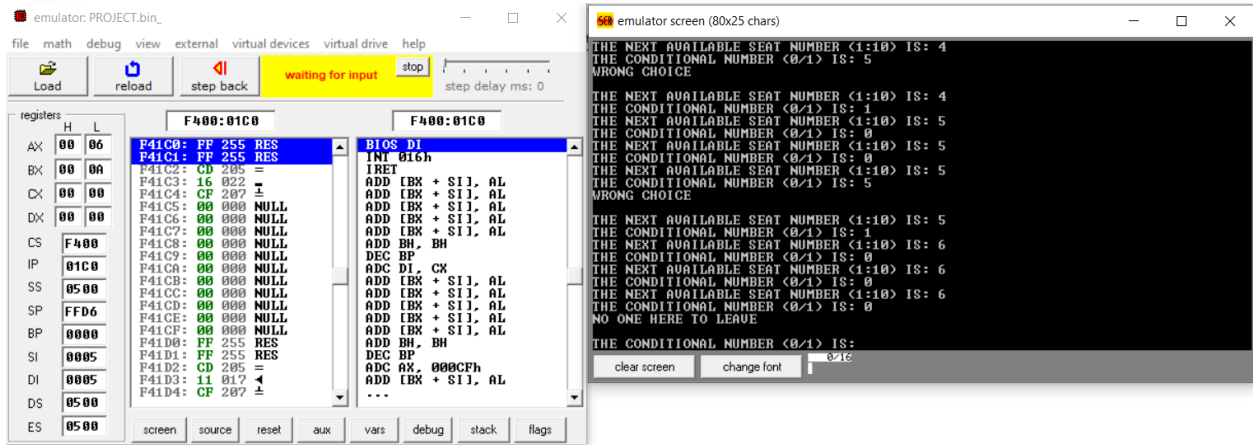
```

0500:0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0500:0010  E2 FA BB 00 00 50 56 EB 3D 45 4E 54 45 52 20 31
0500:0020  20 49 46 20 53 4F 4D 45 4F 4E 45 20 45 4E 54 45
0500:0030  52 45 44 20 54 48 45 20 42 41 4E 4B 20 41 4E 44
0500:0040  20 30 20 49 46 20 53 4F 4D 45 4F 4E 45 20 4C 45
0500:0050  46 54 20 00 00 BE 19 00 2E 8A 04 3C 00 74 07
0500:0060  46 B4 0E CD 10 EB F2 5E 58 50 56 EB 3C 23 23 23
  
```

The emulator screen (80x25 chars) shows the following text:

```

THE CONDITIONAL NUMBER IS: 1
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 3
THE CONDITIONAL NUMBER IS: 0
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 3
THE CONDITIONAL NUMBER IS: 0
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 3
THE CONDITIONAL NUMBER IS: 0
NO ONE HERE TO LEAVE
THE CONDITIONAL NUMBER IS: 1
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 4
THE CONDITIONAL NUMBER IS: 1
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 5
THE CONDITIONAL NUMBER IS: 0
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 5
THE CONDITIONAL NUMBER IS: 0
NO ONE HERE TO LEAVE
THE CONDITIONAL NUMBER IS: 7
WRONG CHOICE
THE NEXT AVAILABLE SEAT NUMBER <1:10> IS: 5
THE CONDITIONAL NUMBER IS: 0
  
```

All the simulations went perfect.