

Micro-Processors 1 Project

Omar Nashat Shahd Abdelrhman Mark Milad
Mariam Ahmad Ziad Hazem

Contents

Macros:	2
PrintMsg(str):	2
Screen1(str1, str2, str3, separator):	2
ChatScreen(name1, name2, separator):	2
GetNm(namein):	2
GameScreenSplit(separator):	3
Prntreg(reg,regname):	3
Procedures:	4

Macros:

PrintMsg(str):

Moves cursor to the position in Dx.

```
mov ah,2  
int 10h
```

Prints the value of the input variable using (int 21h/9).

```
mov ah,9  
mov dx,offset str  
int 21h
```

Screen1(str1, str2, str3, separator):

Displays the first menu (the input variables are the messages that are shown) using **PrintMsg**.

```
mov dx,0818H  
PrintMsg str1
```

ChatScreen(name1, name2, separator):

Clears the screen.

```
mov ax,3  
int 10h
```

Then displays the name of the players and format the screen for chatting using **PrintMsg**.

```
mov dx,0102h  
PrintMsg Name1
```

GetNm(namein):

Checks for the first letter of the name to validate that it is a letter by reading the character with out echo (int 21h/7).

```
mov ah,7  
int 21h
```

Then it performs necessary checks to validate it is a letter before displaying the letter on the screen or waiting for another input.

```
mov ah,2  
mov dl,al  
int 21h
```

Then the rest of the name is read by (int 21/0A).

```

mov ah,0Ah
mov dx,offset namein
int 21h

```

Then shifts the whole name array by one place using movsb.

```

std
rep movsb
cld

```

Finally it adds the first character and increments the name size.

```

mov namein+2,d1

inc namein+1

```

GameScreenSplit(separator):

Splits the lower part of the screen to be used as chat using PrintMsg.

```

mov dx,1600H
PrintMsg separator

```

Splits the screen vertically by printing '|' using (int 21h/2) in a loop.

```

mov cx,16h
mov ah,2
Vsep:  mov dh,c1
      dec dh
mov dl,27h
int 10h
mov dl,'|'
int 21h
loop Vsep

```

Prntreg(reg,regname):

Prints the register name by using the regname variable with the offset in DI the prints it by using (int 21h/9).

```

mov ah,9
mov dx,offset regname
add dx,di
int 21h

```

Then prints the register value using the reg variable with the offset in SI using same method as regname.

Finally it sets the cursor position to the next line using (int 10h/2).

```
add ch,2
mov dx,cx
mov bh,0
mov ah,2
int 10h
```

Procedures: