**'union REGS'**

**'union REGS'** is just an union variable declared in DOS.H ( you can very well have a reference over this dos.h header file ) would be defined like this   
  
union REGS {   
 struct WORDREGS x;  
 struct BYTEREGS h;  
};  
  
in addition definition for both WORDREGS and BYTEREGS can be found in DOS.H.   
  
Why these data structures? this is just a variable similar to common structure and union variable that we used to define.  
  
Whats special with this ?   
 Not much ! but its defined in such a way duplicating the Register structure(Actual 8086 CPU Register Structure )   
  
What about this int86() ???   
  
 This is a function defined in dos.h ,out of which you can generate interrupts to an x86 based CPU it can be read as 'interrupt86' it takes up three arguments   
int86(int interruptNo , union REGS \*ireg , union REGS \*oreg);  
  
here interruptNo referes the interrupt number (Common set of functions will be grouped into interruptNo , eg: All video services will hold the interruptNo 0x10 and so on ), before generating interrupt to the CPU you must provide all the essential data's regarding the service you request   
  
eg:  
  
typedef char BYTE;  
BYTE GetVideoMode( void )  
{  
union REGS inregs, outregs;  
inregs.h.ah = 0x0F;  
int86( 0x10, &inregs, &outregs );  
return( outregs.h.al );  
}  
  
the function given above retrieves the current Video mode of the system using interrupt no 0x10 (as i mentioned earlier all the service request for video devices will hold interrupt no 0x10)  
interrupt no 0x10 is alone not sufficient for the CPU to understand your request in addition you have to specify the sub function 0x0F ( means request video mode ) to the appropriate register variable ( here .h.ah = 0x0f; ).  
  
now once the  function get called it will first deduct the interrupt type ( 0x10 - Video services ) and further sub Function type (0x0f request view mode )and generate the CPU interrupt , after processing the interrupt CPU will fix the results to the register set , which will be populated to &outregs ( make sure &outregs will hold the results for current operation as per your scenario if the ah register holds 0 means successful and failure if 1 )

Printf ("The row position of cursor on selected page - %d",regs.h.ch);

printf(" The column position of cursor on selected page - %d",regs.h.cl);