In the name of god

Hi there,

My project consists of nine c++ source files and a header file.

CPP:

Inst1 : instructions first to 14(the beginning of LOAD family).
Inst2 : LOAD family.
Inst3 : STORE family.
Inst4 : Stack related tasks such pop and etc(after STORE family until Swap)
Inst5 : Math family.
Inst6 : Cast family.
Inst7 : Condition family.
JVM : the class that will solve the problem.
Main : ;)

H:

JVM : Class definition of JVM.

All the instructions were described clear in the class so it seems that extra descriptions are not necessary for friend functions(Commands).

JVM.H:

Prototype of the object that solve the problem(JVM). It has its implementations in JVM.CPP and friend functions in Inst1…7.

-short sp

My stack pointer.

-short pc

My program conter.

-char wide

Determine if previous command is wide prefix.

-int wherethemem

Determine that where the memory must be shown in table.

-FILE * mystream
The CLS file.

-void * loadedfile
Load CLS file in RAM by mean of this pointer.

-pointofunc * alljobs
All the friend functions(Commands) place in this array.

Main.CPP:

Get the command line parameters and declare an object to solve the problem and go on until the return value of file become one specific value(path[0] == 0).

JVM.CPP:

Implementation of JVM class.

-JVM(char * const path,int & phase)
Do all things : Initialize fields, load file, get key to determine what must be done, get the next command of file and call its function and prepare for next file if we choose "O". phase determine if it's called from command line or not.

-~JVM()
My vacuum cleaner, clear screen show cursor set default color and use finalize to deal with all the dynamic fields.

-static void between(int howlong,void * fromwhere,void * towhere)
For moving between two memory places.

-static void revbetw(int howlong,void * fromwhere,void * towhere)
For converting big endian to little endian and vice versa.

-void wherepath(char * const path)
Gets the path of input CLS file.

-void preint()
Draws static things on the screen such as line and etc.

-void getint(int & what)
Gets a number that must set **wherethemem.**

-char sure()

For dangerous choices I make it to make confirmation available.

-JVM(JVM & cockroach)
Private copy constructor for avoid copying.

-void setinstructions()
Set **alljobs** to friend functions.

-char whosnext(int go)
Get the next byte of loadedfile and depending on go add one to pc or not.

-void finalize()
Close all files Delete all pointers and etc.

-void pusher(int howlong,void * what)
Push from somewhere on the top of stack howlong bytes.

-void poper(int howlong,void * what)
Pop from the top of stack to somewhere howlong bytes.

-void messenger(const char * mycommand,const short * index,const char * message,int size = 1)
Update screen with changeable things.

-void memmanager()
drawtable uses it to draw the memory table.

-void drawtable(int where)
Draw the memory table.

Inst1:

-int count
Counter of some loops.

Inst5:

-void longnegative(long * tmp)
Make one long number negative because LDIV & LMUL & LADD & LSUB need it I make it as function.

-void longadd(long * tmp1,long * tmp2)
Add two longs both LSUB & LADD use it.

-void longsub(long * tmp1,long * tmp2)
Subtract two longs both LSUB & LADD use it.

NOTE: You must set *Options→Directories* to right directory.

I've test each of commands individually. You can find commands in n.cls file in JVM directory after bit 336'Th.

I also have test file named In.cls in that directory, its job is to print Fibonacci serie and determine which number doesn't have any divisor by printing -1 after it(with invoke interface).

I also prepare you exe file to use it if any unpredicted conditions take place.

Naeem Esfahani : 810180158

Saturday, January 25, 2003
Javirma project description for assembly course

}}}------>Naeem