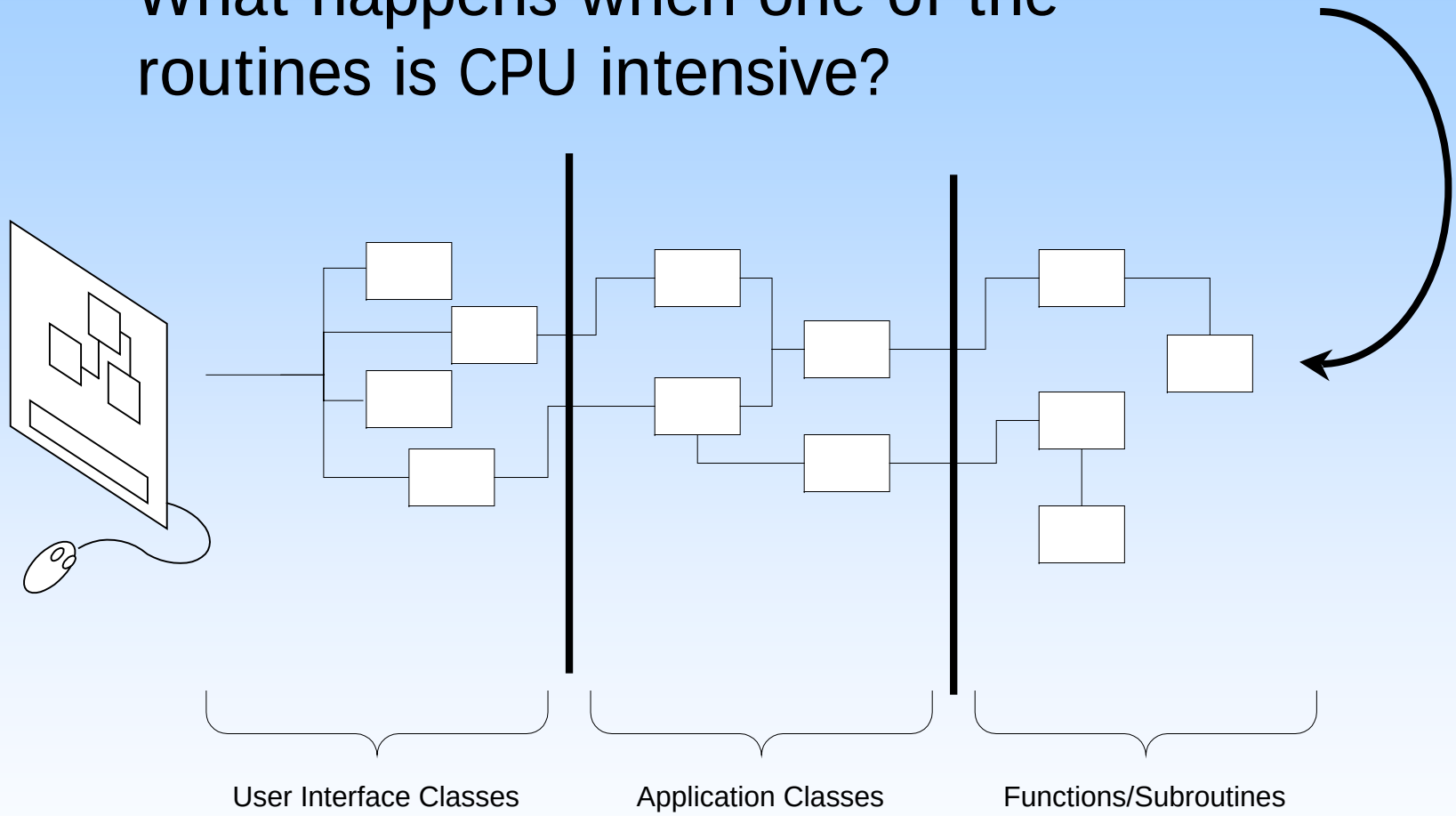
Math 157

Threads (why is my interface freezing?)

March 11, 2009

Scientific/Technical Applications :

What happens when one of the routines is CPU intensive?



A sample computationally intensive task

Multiply two matrices together --- C = A*B

```
for(i = 0; i < N; i++)
{
for(j = 0; j < N; j++)
{
      C[i][j] = 0.0;
      for(k = 0; k < N; k++)
      {
       C[i][j] += A[i][k]*B[k][j];
      }
}}
```

A triple do loop … N^3 operations.

If N = 200, then $8x10^6$ operations are required.

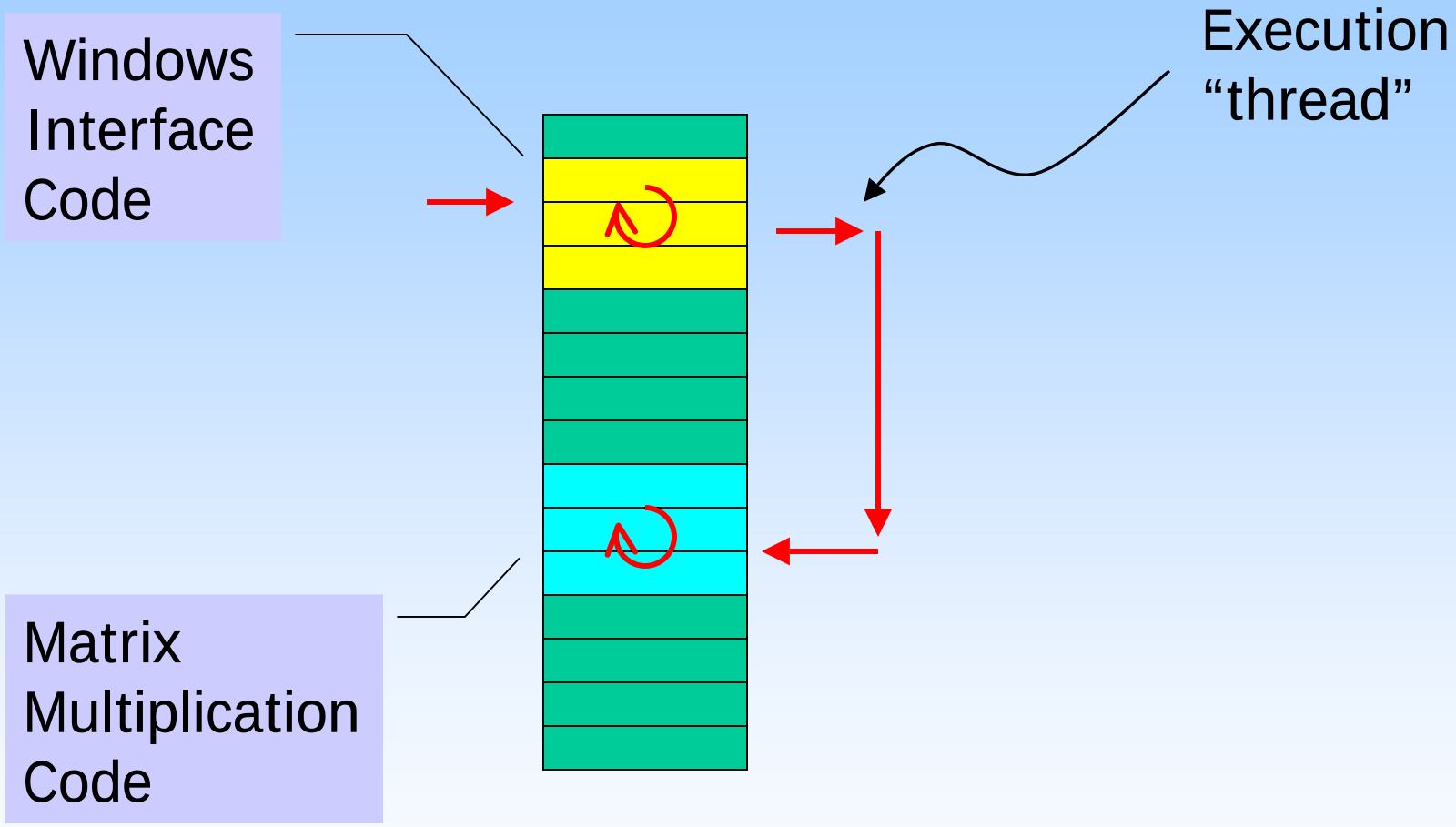
What happens if the calculation is done in a console project?

While the CPU intensive calculation is running, standard user input is not processed ...

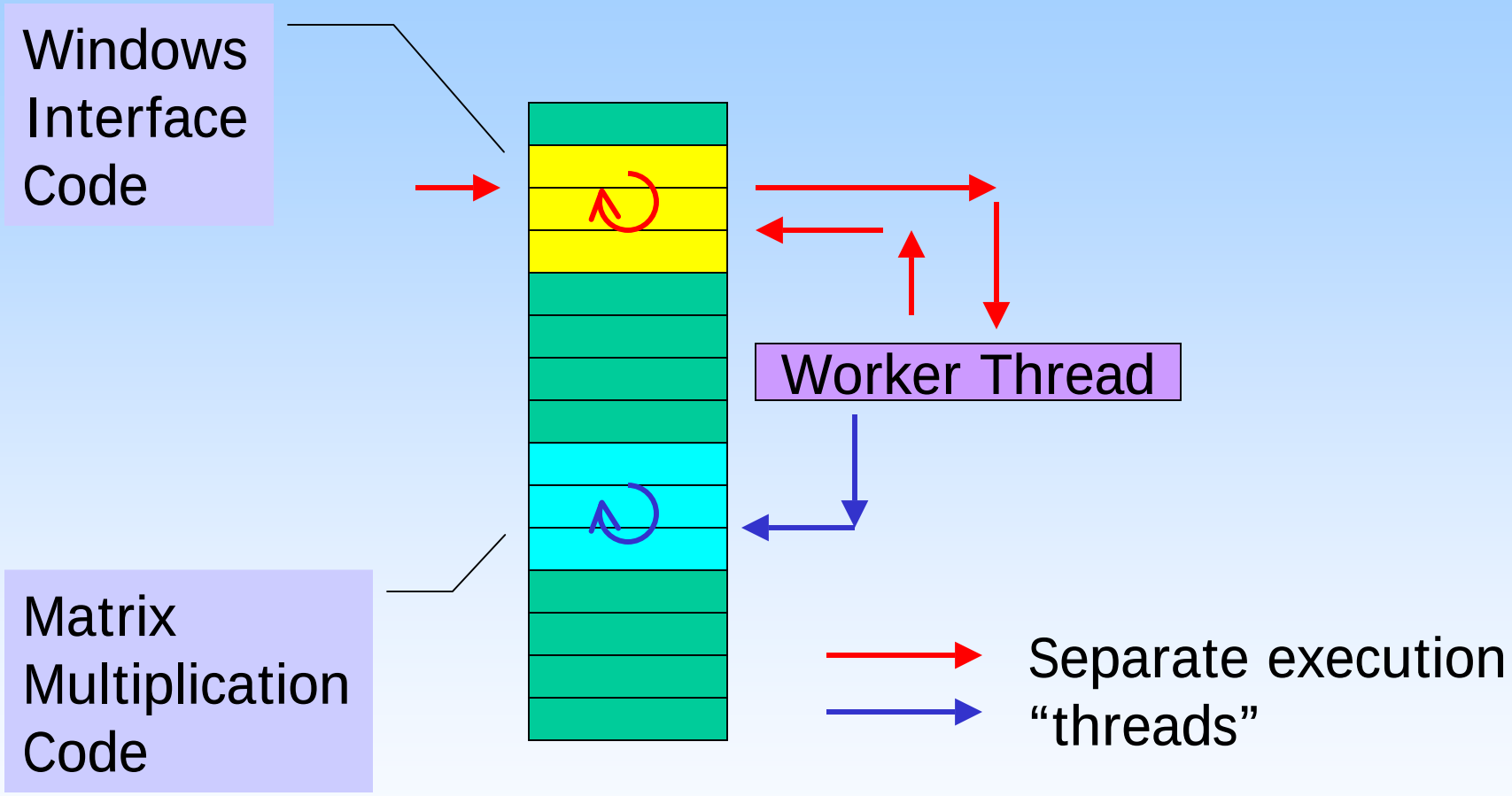
What happens if the calculation is done in a windows application?

The interface “freezes” when the CPU intensive computation is being done.

What's the problem?

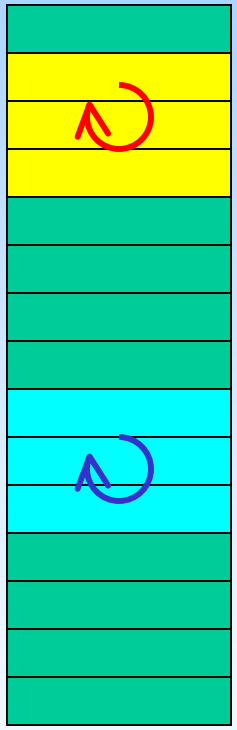


What's the solution?



Multi-threaded execution

Windows Interface Code

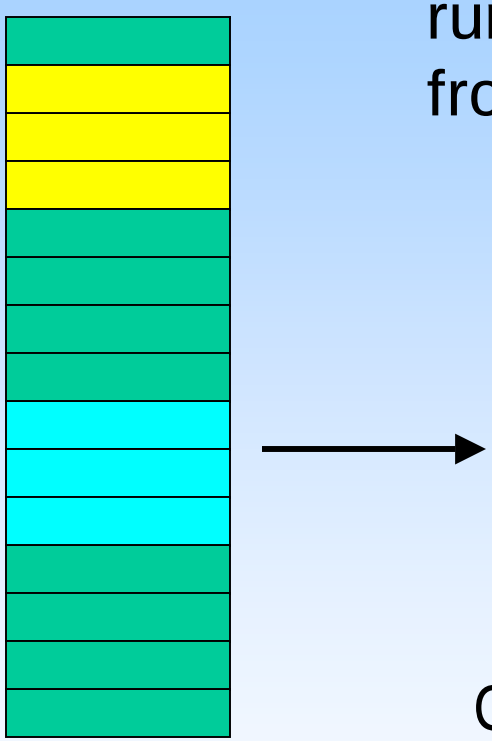


Matrix Multiplication Code

Separate execution “threads”

CPU activity is split between the “threads”

How does one implement this (Qt) ?

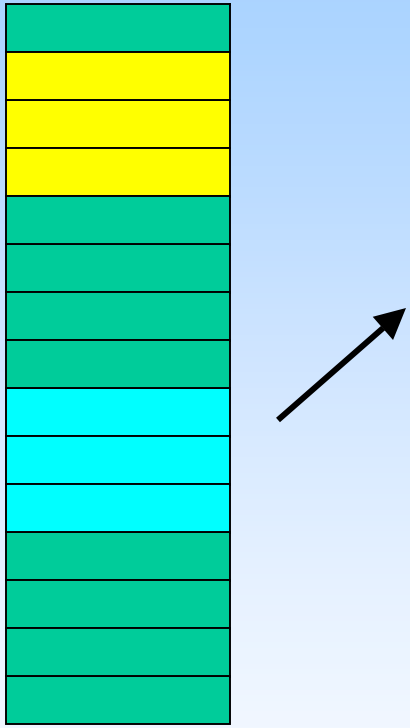


Encapsulate the CPU intensive code in the run() member function of a class derived from **QThread**

```
void  run()
{
    ****
}
```

Call the start() method inherited from **Qthread**

How does one implement this (MFC windows)?



Encapsulate the CPU intensive code in a routine of the form

```
UINT routineName( LPVOID pParam )
{
    ****
}
```

Call the MFC routine

AfxBeginThread(*routineName*, ...);

Details?

You can set the priority of the thread.

Important! You set the priority of the CPU intensive component **lower** than the interface component!

Communication between program units executing under multiple threads is complicated...

- Be careful that identical data is not being accessed simultaneously...
- Threads can dead-lock, each waiting for the other to do something...
- Stopping a running thread requires careful programming...

Advice…

Qt and MFC help examples are ok for simple multi-threaded programming.

More complicated programming? Find a good book.

To “play” around -- JAVA is nice because threads are part of the standard packages; e.g. java.lang.Threads.