Module 7 (Lectures 32-33) Program profiling

1. Consider the problem of multiplying two 1024x1024 double float matrices. This can be done using the triply nested *for* loops that you are familiar with from a course on matrices. Implement a program for this matrix multiplication; the values of the matrix elements are unimportant, so you can initialize them to contain arbitrary values. Structure your program to be made up of functions

   MatInit( ) which initializes the matrices
   MatMult( ) which does the matrix multiplication
   main( ) which calls MatInit() and MatMult()

   We are interested here in measuring the amount of time spent in each execution of the innermost loop. Try and do this using (i) gettimeofday( ), (ii) the *rdtsc* instruction on a system using an Intel© processor.
2. Now consider the use of the function level profiling mechanism. You can do this using the –pg option of gcc, and the gprof program. Compile your matrix multiplication program using gcc –pg. Execute it and then study the function level profile using gprof. What is the estimate of time spent executing in the function MatMult( )?
3. Next attempt to measure the total execution time in function MatMult( ) using the mechanisms that you experimented with in question 6, *viz*, gettimeofday( ) and *rdtsc*. Instrument your program with calls to these before and after the call to function MatMult( ) in main( ). How do these measurements compare with the estimate obtained from question 7?
4. Repeat question 7 under both heavy and light load system load conditions. To create light load conditions on your system, make sure that no other programs are running when you are doing the profiling run. To create heavy load conditions, have several other programs in execution while you are doing the profiling run.
5. The function level profiling mechanism used in prof and gprof uses PC sampling to generate execution time estimates. Try and generate a program (as suggested in the lecture) which is made up of 2 functions, A( ) and B( ), where the bulk of execution time is actually spent in function A( ), but the PC sampling leads to the conclusion that B( ) is the function where most of the execution time is spent.