Imperial College London
Department of Computing

Computer Systems (M2)
Pentium programming – lab exercise

You'll learn more about assembly language programming as the course progresses. In the meantime you're encouraged to run the following program. To do so, you'll need to login to one of the CSG machines running **Linux** (not Windows!). You can do this exercise individually or with others. It shouldn't take more than 40 minutes to complete. Have fun!!

* Type the program in the box below into a file called `hello.s` **Do not make any mistakes!!!!**

```
segment .data                       switch to data segment
msg     db 'Hello world!',0xA       declare and initialise variable msg
len     equ $-msg                   set constant len = number of bytes in msg

segment .text                       switch to text (i.e. code) segment
global _start                       make _start visible outside of this file

_start:                             program starts here
        mov  eax, 5                 number of times to repeat outer loop
outer:
        mov  ebx, 1000000000        repeat inner loop 1 billion times. type 1 followed
                                        by 9 zeros - do not type any more zeros!
inner:
        dec  ebx                    execute this & next instruction 1 billion times
        jg   inner                  jump if ebx greater than zero to label 'inner'
        dec  eax                    decrement eax outer loop counter
        jg   outer                  jump if eax greater than zero to label 'outer'

        mov  eax, 4                 linux system call 4, i.e. write ()
        mov  ebx, 1                 file descriptor 1, i.e. standard output
        mov  ecx, msg               address of variable 'msg'
        mov  edx, len               number of bytes in message to write
        int  0x80                   interrupt Linux, i.e. Linux will write the message

        mov  eax, 1                 linux system call 1  i.e. exit ()
        mov  ebx, 0                 error code 0, i.e. no errors
        int  0x80                   interrupt Linux, i.e. Linux will exit the program
```

*	Assemble into an object file version with: `nasm -f elf hello.s`	`nasm` is the Netwide assembler. The command will produce an object file named `hello.o` if there are no errors in file `hello.s`
*	Then link into an executable program with: `ld -s -o hello hello.o`	`ld` is the Linux object file 'linker' which can (amongst other things) link several object files into one executable program.
	Run the program with: `hello` or `./hello`	The program executes over 10 billion Pentium instructions! 5 billion `dec` instructions and 5 billion `jg` instructions.
*	Find the size of the executable file with: `wc -c hello`	This is probably the smallest complete program you'll ever create during your degree!
*	Find the size of the code and data with: `size hello` Why is there a difference between the sum of these sizes and the size of the executable program file?	The code size is given in the 'text' column! Try the command `file hello` for fun also.
*	Run and time the program with: `/usr/bin/time -p hello` If you run the program several times, the times may differ. Why?	For a more verbose timing output use the command: `/usr/bin/time -v hello` View the cpu type using `more /proc/cpuinfo`
*	Login to other CSG Linux machines with different processors and compare the speed.	For names of machines, go to the csg home page and click on Computer List on the panel. Try older/slower machines